



Gestión de espacios y eventos culturales (GEEC)

Proyecto de fin de carrera

Versión: 1.3

Fecha: 17/05/2013

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

HOJA DE CONTROL

Proyecto	Gestión de espacios y eventos culturales (GEEC)		
Entregable	Proyecto de fin de carrera		
Nombre del fichero	Memoria Proyecto Fin de Carrera.odt		
Autor	Christian D. Jiménez García		
Versión	1.3	Fecha Versión	17/05/2013
Aprobado por		Fecha Aprobación	
		Nº de páginas	248

REGISTRO DE CAMBIOS

Versión	Causa del cambio	Responsable de cambio	Fecha
1.0	Documento inicial del proyecto.	Christian D. Jiménez García	12/07/2011
1.1	Continuación del proyecto.	Christian D. Jiménez García	15/08/2012
1.2	Versión definitiva inicial.	Christian D. Jiménez García	18/01/2013
1.3	Cambio en la enumeración de la biografía. Añadido el Anexo de manual de instalación. Añadido el Anexo de manual de usuario.	Christian D. Jiménez García	12/04/2013

CONTROL DE DISTRIBUCIÓN

Nombre y Apellidos	Cargo	Área	Nº Copias
Miguel Ángel Patricio	Director de proyecto	UC3M	1



Gestión de espacios y eventos culturales (GEEC)

Proyecto de fin de carrera

*A mi familia por su apoyo incondicional,
su ánimo y su paciencia.
A Yoli, Ainara y Noa por el tiempo que
les he robado.
A Ricardo por acompañarme en el viaje
que fue finalmente su PFC.
A Javier González por su colaboración
para la creación y el análisis de la
aplicación.
A Miguel A. Patricio por su paciencia y
guía en la elaboración del proyecto.*



ÍNDICE

1 Introducción.....	11
1.1 Ambientación.....	11
1.2 Motivación.....	12
1.3 Objetivos.....	12
1.4 Medios/Infraestructuras.....	14
1.4.1 Arquitectura cliente/servidor.....	14
1.4.2 Software de cliente: Navegador web.....	15
1.4.3 Software de servidor.....	16
1.4.3.1 Lenguaje de programación.....	16
1.4.3.2 Servidor de aplicaciones.....	17
1.4.3.3 Sistema gestor de bases de datos.....	17
1.4.4 Entorno de desarrollo.....	17
1.5 Organización de la memoria.....	18
2 Requisitos.....	19
2.1 Restricciones.....	19
2.2 Suposiciones.....	20
2.3 Requisitos funcionales.....	20
2.3.1 RFUN01: Autenticar usuarios.....	20
2.3.2 RFUN02: Creación de grupos de trabajo.....	20
2.3.3 RFUN03: Listado de grupos de trabajo.....	20
2.3.4 RFUN04: Edición de grupos de trabajo.....	20
2.3.5 RFUN05: Borrado de grupos de trabajo.....	21
2.3.6 RFUN06: Gestión de empleados.....	21
2.3.7 RFUN07: Listado de empleados.....	21
2.3.8 RFUN08: Edición de empleados.....	21
2.3.9 RFUN09: Borrado de empleados.....	21
2.3.10 RFUN10: Creación de usuarios.....	21
2.3.11 RFUN11: Borrado de usuarios.....	21
2.3.12 RFUN12: Creación de perfil de usuario en grupo.....	21
2.3.13 RFUN13: Borrado de perfil de usuario en grupo.....	22
2.3.14 RFUN14: Creación de contactos.....	22
2.3.15 RFUN15: Listado y búsqueda de contactos.....	22
2.3.16 RFUN16: Edición de contactos.....	22
2.3.17 RFUN17: Borrado de contactos.....	22
2.3.18 RFUN18: Creación de espacios.....	22
2.3.19 RFUN19: Listado de espacios.....	22
2.3.20 RFUN20: Edición de espacios.....	22
2.3.21 RFUN21: Borrado de espacios.....	23



2.3.22 RFUN22: Creación de eventos.....	23
2.3.23 RFUN23: Listado de eventos.....	23
2.3.24 RFUN24: Planificación de eventos.....	23
2.3.25 RFUN25: Edición de eventos.....	23
2.3.26 RFUN26: Autorización o rechazo de eventos.....	23
2.3.27 RFUN27: Borrado de eventos.....	24
2.3.28 RFUN28: Actualización del histórico de eventos.....	24
2.3.29 RFUN29: Histórico de eventos.....	24
2.3.30 RFUN30: Paginación de listados.....	24
2.3.31 RFUN31: Filtros de búsqueda.....	24
2.4 Requisitos no funcionales.....	24
2.4.1 Requisitos de datos.....	25
2.4.1.1 RDAT01: Datos sobre usuarios.....	25
2.4.1.2 RDAT02: Datos de autenticación de usuarios.....	25
2.4.1.3 RDAT03: Datos sobre los grupos.....	25
2.4.1.4 RDAT04: Datos sobre actividad de usuario.....	25
2.4.1.5 RDAT05: Datos sobre contactos y organizaciones.....	25
2.4.1.6 RDAT06: Datos sobre espacios.....	25
2.4.1.7 RDAT07: Datos sobre eventos.....	25
2.4.2 Requisitos de seguridad.....	26
2.4.2.1 RSEG01: Privacidad de usuarios y de datos sensibles almacenados.....	26
2.4.2.2 RSEG02: Existencia de un usuario base o superadministrador.....	26
2.4.2.3 RSEG03: Perfiles de usuario.....	26
2.4.2.4 RSEG04: Acceso mediante perfiles y grupos.....	26
2.4.2.5 RSEG05: Sesiones por usuario.....	26
2.4.3 Requisitos de interfaces de usuario.....	26
2.4.3.1 RIUS01: Internacionalización.....	27
2.4.3.2 RIUS02: Retro-alimentación al usuario.....	27
2.4.3.3 RIUS03: Información de ayuda a los usuarios.....	27
2.4.3.4 RIUS04: Menú de navegación.....	27
2.4.4 Requisitos de interfaces externos.....	27
2.4.4.1 RIEX01: Compatibilidad en navegadores.....	27
2.4.5 Requisitos de interfaces de comunicación.....	27
2.4.5.1 RCOM01: Protocolos de comunicación.....	27
2.5 Requisitos de rendimiento.....	27
2.5.1.1 RREN01: Tiempos de respuesta en peticiones.....	28
2.5.1.2 RREN02: Concurrencia de usuarios.....	28
2.5.1.3 RREN03: Paginación de datos.....	28
2.6 Requisitos de desarrollo.....	28
2.6.1.1 RDES01: Arquitectura de sistema.....	28
2.6.1.2 RDES02: Ciclo de vida.....	28
2.7 Requisitos tecnológicos.....	28



2.8 Atributos del sistema.....	28
2.8.1 Sistema portable.....	29
2.8.2 Evolutivo.....	29
3 Análisis y diseño.....	30
3.1 Análisis de casos de uso.....	30
3.1.1 Autenticación de usuarios.....	32
3.1.1.1 Autenticar usuario.....	32
3.1.1.2 Cerrar sesión.....	33
3.1.2 Gestión de grupos.....	34
3.1.2.1 Crear grupo.....	34
3.1.2.2 Listar grupos.....	35
3.1.2.3 Modificar grupo.....	36
3.1.2.4 Eliminar grupo.....	36
3.1.3 Gestión de usuarios.....	38
3.1.3.1 Crear empleado.....	39
3.1.3.2 Listar empleados.....	40
3.1.3.3 Crear usuario.....	41
3.1.3.4 Modificar usuario.....	41
3.1.3.5 Eliminar usuario.....	42
3.1.3.6 Listar roles de usuario en grupo.....	43
3.1.3.7 Añadir rol de usuario en grupo.....	43
3.1.3.8 Eliminar rol de usuario de grupo.....	44
3.1.3.9 Eliminar todos los roles de usuario de grupo.....	45
3.1.3.10 Modificar empleado.....	46
3.1.3.11 Eliminar empleado.....	47
3.1.4 Gestión de empresas.....	48
3.1.4.1 Crear empresa.....	48
3.1.4.2 Listar empresas.....	49
3.1.4.3 Modificar empresa.....	50
3.1.4.4 Eliminar empresa.....	51
3.1.5 Gestión de contactos.....	52
3.1.5.1 Crear contacto.....	52
3.1.5.2 Listar contactos.....	53
3.1.5.3 Modificar contacto.....	54
3.1.5.4 Eliminar contacto.....	55
3.1.6 Gestión de espacios.....	56
3.1.6.1 Crear espacio.....	56
3.1.6.2 Listar espacio.....	57
3.1.6.3 Modificar espacio.....	58
3.1.6.4 Eliminar espacio.....	59
3.1.7 Gestión de eventos.....	61
3.1.7.1 Agenda de eventos.....	61



Gestión de espacios y eventos culturales (GEEC)

Proyecto de fin de carrera

3.1.7.2 Crear evento.....	62
3.1.7.3 Modificar evento.....	62
3.1.7.4 Listar eventos.....	63
3.1.7.5 Listar del histórico de eventos.....	64
3.1.7.6 Aceptar evento.....	65
3.1.7.7 Rechazar evento.....	66
3.1.7.8 Ejecutar evento.....	67
3.1.7.9 Rechazar evento fuera de fecha.....	67
3.1.7.10 Eliminar evento rechazado.....	68
3.1.7.11 Mover eventos al histórico.....	68
3.2 Modelo de datos.....	69
3.2.1 Modelo conceptual de datos.....	69
3.2.2 Modelo lógico estándar.....	71
3.2.3 Modelo físico.....	73
3.3 Diagramas de clases (estructurales).....	76
3.3.1 Diagrama de clases del modelo de dominio.....	78
3.3.2 Diagrama de clases de controladores.....	79
3.3.3 Diagrama de clases de servicios.....	81
3.3.4 Diagrama de clases de acceso a datos.....	82
3.4 Diagramas de objetos y de secuencia.....	85
3.4.1 Funcionalidades de administración de la aplicación.....	85
3.4.1.1 Diagrama de secuencia de la creación de grupo de trabajo.....	87
3.4.1.2 Diagrama de secuencia del listado de grupos de trabajo.....	88
3.4.1.3 Diagrama de secuencia de la modificación de grupo de trabajo.....	89
3.4.1.4 Diagrama de secuencia del borrado de grupo de trabajo.....	90
3.4.1.5 Diagrama de secuencia de la creación de empleado.....	91
3.4.1.6 Diagrama de secuencia del listado de empleados.....	92
3.4.1.7 Diagrama de secuencia de la modificación de empleado.....	93
3.4.1.8 Diagrama de secuencia del borrado de empleado.....	94
3.4.1.9 Diagrama de secuencia de la creación de usuario.....	95
3.4.1.10 Diagrama de secuencia del borrado de usuario.....	96
3.4.1.11 Diagrama de secuencia de la inserción de rol de usuario en grupo.....	97
3.4.1.12 Diagrama de secuencia del borrado de rol de usuario en grupo.....	98
3.4.2 Funcionalidades de gestión de empresas y contactos.....	99
3.4.2.1 Diagrama de secuencia de la creación empresa.....	101
3.4.2.2 Diagrama de secuencia del listado de empresas.....	102
3.4.2.3 Diagrama de secuencia de la modificación de empresa.....	103
3.4.2.4 Diagrama de secuencia del borrado de empresa.....	104
3.4.2.5 Diagrama de secuencia de la creación de contacto.....	105
3.4.2.6 Diagrama de secuencia del listado de contactos.....	106
3.4.2.7 Diagrama de secuencia de la modificación de contacto.....	107
3.4.2.8 Diagrama de secuencia del borrado de contacto.....	108



3.4.3 Funcionalidades de gestión de espacios y eventos.....	108
3.4.3.1 Diagrama de secuencia de la creación de espacio.....	110
3.4.3.2 Diagrama de secuencia del listado de espacios.....	111
3.4.3.3 Diagrama de secuencia de la modificación de espacio.....	112
3.4.3.4 Diagrama de secuencia del borrado de espacio.....	113
3.4.3.5 Diagrama de secuencia de la agenda de eventos.....	114
3.4.3.6 Diagrama de secuencia de la creación de evento.....	115
3.4.3.7 Diagrama de secuencia de la modificación de evento.....	116
3.4.3.8 Diagrama de secuencia del listado de eventos.....	117
3.4.3.9 Diagrama de secuencia de aceptación de evento.....	118
3.4.3.10 Diagrama de secuencia de rechazo de evento.....	119
3.4.3.11 Diagrama de secuencia del listado del histórico de eventos.....	120
3.4.3.12 Diagrama de secuencia del paso de eventos a ejecutados.....	121
3.4.3.13 Diagrama de secuencia del paso de eventos a rechazados.....	121
3.4.3.14 Diagrama de secuencia del paso de eventos al histórico.....	122
3.4.3.15 Diagrama de secuencia del borrado de eventos rechazados.....	123
4 Tecnologías utilizadas.....	124
4.1 Frameworks utilizados.....	124
4.1.1 JSF (JavaServer Faces) y componentes Primefaces.....	126
4.1.2 Spring.....	126
4.1.3 Spring Security.....	128
4.1.4 JPA e Hibernate.....	129
4.1.5 JSON y Jackson.....	130
4.1.6 SLF4J y Log4Java.....	131
4.1.7 Junit y EasyMock.....	131
4.2 Entorno de desarrollo.....	132
4.2.1 Eclipse IDE.....	132
4.2.2 Apache Tomcat.....	132
4.2.3 Sistema gestor de base de datos MySQL.....	133
4.2.4 Apache Maven.....	133
5 Conclusiones y futuros trabajos.....	134
5.1 Objetivos (cumplidos).....	134
5.2 Futuros trabajos.....	135
6 ANEXO A. Scripts de base de datos.....	138
6.1 Script de creación de base de datos.....	138
6.2 Script de inicialización de la tabla de usuario.....	145
6.3 Script de inicialización de la tabla rol.....	145
6.4 Script inicialización de la tabla de usuario_rol.....	145
6.5 Script inicialización de la tabla de actividad.....	145
6.6 Script inicialización de la tabla de comunidad.....	145
6.7 Script de inicialización de la tabla provincia.....	146
6.8 Script de inicialización de la tabla municipio.....	147



6.9 Script de inicialización de la tabla estado.....	147
6.10 Script de inicialización de la tabla tipo_entidad.....	147
6.11 Script de inicialización de la tabla tipo_persona.....	148
7 ANEXO B. Manual de instalación.....	149
7.1 Instalación de la Máquina Virtual Java 6.....	149
7.2 Instalación del SGBD MySQL 5.....	154
7.3 Creación del esquema de base de datos y ejecución de los scripts de inicialización.....	165
7.4 Instalación del Contenedor Servlet Apache Tomcat 6.....	172
7.5 Despliegue y configuración de la aplicación.....	178
8 ANEXO C. Manual de usuario.....	182
8.1 Acceso al sistema.....	182
8.2 Administración del sistema.....	186
8.2.1 Gestión de grupos.....	186
8.2.1.1 Creación de un nuevo grupo de trabajo.....	188
8.2.1.2 Edición de un grupo de trabajo.....	188
8.2.1.3 Borrado de un grupo de trabajo.....	190
8.2.1.4 Búsqueda o filtro de grupos de trabajo.....	190
8.2.2 Gestión de usuarios.....	191
8.2.2.1 Creación de un nuevo empleado.....	192
8.2.2.2 Edición de un empleado.....	193
8.2.2.3 Borrado de un empleado.....	194
8.2.2.4 Búsqueda o filtro de empleados.....	195
8.2.2.5 Creación de usuario.....	195
8.2.2.6 Modificación de usuario.....	196
8.2.2.7 Borrado de usuario.....	197
8.2.3 Gestión de roles en grupos.....	197
8.2.3.1 Creación de perfil en grupo.....	199
8.2.3.2 Añadir perfil en grupo.....	200
8.2.3.3 Eliminar perfil de grupo.....	201
8.2.3.4 Eliminar grupo y perfiles.....	201
8.2.4 Gestión de empresas.....	202
8.2.4.1 Creación de una nueva empresa.....	203
8.2.4.2 Edición de empresa.....	204
8.2.4.3 Borrado de empresa.....	205
8.2.4.4 Búsqueda o filtro de empresas.....	206
8.2.5 Gestión de contactos.....	206
8.2.5.1 Creación de un nuevo contacto.....	208
8.2.5.2 Edición de un contacto.....	209
8.2.5.3 Borrado de un contacto.....	210
8.2.5.4 Búsqueda o filtro de contactos.....	211
8.2.6 Gestión de espacios culturales.....	211
8.2.6.1 Creación de un nuevo espacio cultural.....	213

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

8.2.6.2 Edición de un espacio cultural.....	237
8.2.6.3 Borrado de un espacio cultural.....	238
8.2.6.4 Búsqueda o filtro de espacios.....	239
8.2.7 Agenda de eventos.....	239
8.2.7.1 Creación de evento sobre la agenda.....	241
8.2.7.2 Modificación de evento sobre la agenda.....	242
8.2.8 Listado de eventos.....	243
8.2.9 Histórico de eventos.....	245
9 ANEXO D. Bibliografía.....	247



1 Introducción

1.1 – Ambientación

1.4 – Medios/Infraestructuras

1.2 – Motivación

1.5 – Organización de la memoria

1.3 – Objetivos

Este documento contiene la memoria del proyecto de fin de carrera GEEC, el cual consistirá en la implementación de un sistema de Gestión de Eventos y Espacios Culturales, el cual se enmarca entre la gestión de espacios escénicos y la ingeniería del software. En este primer capítulo, se desarrollarán diversos puntos que nos permitirán entender el contexto del sistema a desarrollar. En primer lugar se describen las principales razones y motivaciones que han llevado a desarrollar este proyecto. A continuación se enumerarán los medios e infraestructuras que se han utilizado para la creación del sistema. Por último, se realizará una descripción de cómo se ha organizado el resto de la memoria.

1.1 Ambientación

En el mundo de la organización de espacios y eventos culturales, ya sea en el ámbito privado o en el ámbito de la administración pública, y en particular a nivel de los Ayuntamientos de los distintos municipios, no existe una gestión informatizada de los eventos y los espacios culturales de los que se dispone (teatros, salas, espacios al aire libre, ...).

Debido a la ausencia de herramientas específicas para este fin, a la hora de gestionar la programación cultural surgen bastantes problemas derivados de la utilización de herramientas poco apropiadas como listados de papel, hojas de cálculo, etc. Tratamos de no utilizar herramientas centralizadas que provoquen, o puedan provocar, la inconsistencia de datos, falta de sincronización entre usuarios, la imposibilidad de obtener información en tiempo real, etc.

Aunque existen algunas herramientas para este fin, estas son escasas, bastante específicas, la mayoría de tipo comercial y no se ajustan a los requisitos ni a la flexibilidad necesaria para este tipo de gestión.

Una herramienta que alcanzase el nivel de flexibilidad necesario y que permitiese adaptarse a la gestión tanto pública como privada, podría encontrar y copar un segmento de mercado (ya sea a nivel libre como a nivel comercial) que hasta el momento no ha captado la atención de los grupos o empresas de desarrollo pero que, cada vez está adquiriendo más importancia debido al aumento de la oferta y



demanda de este tipo de eventos que está surgiendo, sobre todo para el público infantil, aunque se está produciendo en general a todos los niveles.

1.2 Motivación

La idea de desarrollar este sistema surge tras varios años de experiencia trabajando para distintos organismos tanto privados como públicos que, entre otras tareas, desempeñaban la labor de gestionar espacios dedicados a eventos culturales u organizativos, así como la planificación de actividades en los mismos.

En concreto, se pretende realizar el análisis detallado de las necesidades de un entorno de gestión de espacios culturales u organizativos y crear un sistema que permita cubrir las necesidades de la gestión de espacios y eventos culturales y que sea lo suficientemente flexible para adaptarse al ámbito público y al privado, partiendo de la base del conocimiento acumulado sobre dicha área.

Además, se pretende aportar el conocimiento de las necesidades del usuario el cual disponemos de primera mano basándonos en nuestra experiencia en el campo que se va a estudiar. Afortunadamente, para el desarrollo del sistema de información se han podido realizar reuniones con distintas personas encargadas de la gestión cultural, pudiendo tener acceso a las necesidades más inmediatas e importantes que se requieren para un sistema de este tipo.

Así mismo, evaluaremos distintas herramientas para la implementación del mismo, las cuales intentaremos que en su mayor parte sean de código abierto, y basarnos en estándares de la industria web para crear una arquitectura que se pueda reutilizar en siguientes evoluciones o nuevos proyectos.

1.3 Objetivos

El principal objetivo del proyecto es elaborar un sistema informático que permita gestionar los espacios y los eventos culturales que puedan interesar a los distintos tipos de entidades que se encargan de planificar u organizar estos aspectos. Dicho sistema debe cumplir los siguientes requisitos básicos:

1. Estudio y análisis del sistema de gestión de espacios y eventos culturales.
2. Estudio de la tecnología utilizada para la implementación de una arquitectura base, y que permita la implementación rápida, sencilla y basada en estándares del desarrollo web que estén ampliamente extendidos, lo cual permitirá encontrar documentación suficiente en caso de ser necesario. Evaluación de las ventajas e inconvenientes y justificación de la elección de los frameworks de desarrollo utilizados para la implementación del proyecto.
3. Análisis y diseño de los módulos necesarios para la gestión de espacios y eventos culturales.



A nivel del sistema a generar sobre el documento que estamos desarrollando se pretenden alcanzar los siguientes objetivos:

1. Permitir la administración y control de los distintos grupos dentro de la organización que se encargan de gestionar la información. Esto permitirá que, si la empresa u organismo está dividida en varias secciones o se encuentra repartida entre distintas ubicaciones físicas, pueda haber distintos grupos que accedan a la aplicación de forma centralizada para gestionar sus datos y coordinar la oferta cultural, e incluso colaborar entre ellos compartiendo información común.
2. Gestionar la información de los espacios para los cuales se quiere realizar la programación de eventos. Además, debe permitir que la información relativa a estos espacios pueda ser completada progresivamente según interese. Es decir, no es obligatorio que la información de los espacios sea completa, pero se podrá ir ampliando según se obtenga la información o a medida que vaya siendo proporcionada por la entidad o personas que gestionen el mismo, aunque realmente no será obligatorio tener esta información para que el sistema funcione correctamente.
3. Planificar y diseñar los eventos que tendrán lugar en los distintos espacios administrados por la aplicación, así como toda la información de utilidad para los mismos, permitiendo realizar un seguimiento y control suficientemente exhaustivo de las mismas.
4. Resolver la problemática que se plantea a la hora de almacenar los datos específicos de los espacios culturales. Es decir, debido a la diversidad y variabilidad de los datos que describen un determinado espacio cultural (una sala de conciertos, un teatro, un salón de actos, ...), no es óptimo diseñar un almacenamiento tabular en base de datos, ya que en muchos casos, la mayoría de los campos pueden quedar en blanco perdiendo rendimiento de base de datos. Habrá que buscar una forma de solventar esta problemática.
5. Mantener un control de contactos compartido entre todos los grupos que participen en la gestión cultural, el cual puede ser interesante a la hora de programar actividades.
6. Crear un sistema de interfaces sencillo e intuitivo que permita a usuarios de distintos niveles aprender a utilizar la aplicación sin precisar de conocimientos especializados a parte de la navegación web.
7. Permitir el acceso a la aplicación desde cualquier punto con conexión a Internet. Debido a que la aplicación ha de estar preparada para gestionar entidades que pueden estar distribuidas en distintos lugares físicos, se debe poder acceder a la misma mediante Internet, así como asegurar la seguridad de los usuarios debido a la sensibilidad de algunos de los datos que se van a manejar.
8. Preparar el sistema para la localización de la interfaz de usuario, ya que puede interesar,



específicamente en el ámbito de organismos públicos, la selección de distintas lenguas en función de la ubicación física donde se utilice el sistema.

9. Como último objetivo, estableceremos el control y organización del desarrollo de la aplicación para permitir la futura ampliación de la funcionalidad de la misma en aspectos relativos a nuevos usuarios, los cuales podrían acceder a consultar información descriptiva sobre los espacios y eventos gestionados por el sistema.

1.4 Medios/Infraestructuras

El objetivo de este apartado será analizar y evaluar las herramientas empleadas y si son adecuadas a la hora de construir una herramienta de gestión de espacios y eventos en la que diversas personas de un equipo de desarrollo deberán colaborar en su implementación de la forma más sencilla posible.

La aplicación a diseñar estará orientada a su utilización a través de una red y en concreto a través de Internet, ya que es interesante que los usuarios puedan acceder a la misma desde distintas ubicaciones. Este motivo nos ha llevado a analizar una serie de herramientas y tecnologías determinadas para el desarrollo del sistema, las cuales pasaremos a detallar a continuación.

1.4.1 Arquitectura cliente/servidor

La arquitectura cliente/servidor se puede definir como una arquitectura distribuida que permitirá a los usuarios finales obtener acceso a la información de forma transparente incluso en entornos multiplataforma. El funcionamiento de la arquitectura cliente/servidor se basa en un proceso cliente que iniciará la petición, un proceso servidor que esperará a que lleguen peticiones de servicio y un software intermedio que proporcionará la interfaz de conectividad entre el cliente y el servidor para poder intercambiar los mensajes.

Este tipo de arquitectura se puede clasificar además según los conceptos Fat client/Thin server o Thin client/Fat server, en función de la parte de la arquitectura donde se quiera que se encuentre el peso de la aplicación. En nuestro caso, elegiremos el segundo tipo ya que presenta una mayor flexibilidad a la hora de desarrollar aplicaciones.

Además la arquitectura cliente/servidor también se puede clasificar en función de capas o tamaño de componentes, es decir, en función de la proporción en que se asignarán las funcionalidades de la aplicación a cada una de las partes. En general se utilizan dos modelos distintos para definir la arquitectura, el modelo de dos capas o Modelo 1 y el modelo de n-capas o Modelo 2. En la actualidad, el modelo más extendido es el Modelo 2, ya que el Modelo 1 aunque resulta muy fácil de programar no es escalable y mueve gran cantidad de código en las páginas finales lo que representa un problema para los diseñadores, los cuales no tienen por qué tener conocimientos de programación. Además el código



de acceso a datos está fuertemente acoplado, por lo que un cambio en el sistema gestor de base de datos implica tener que modificar la aplicación al completo. En nuestro caso, hemos elegido seleccionar el Modelo 2. Este modelo permitirá dividir el código de servidor al menos en tres capas mejorando el rendimiento, proporcionando una mayor flexibilidad de desarrollo, escalabilidad horizontal y vertical, independencia del código con respecto a la representación de datos para el usuario final, desacoplamiento entre el código de negocio y los datos y mejora de la portabilidad de aplicaciones. En el sistema a implementar, se dividirá la parte servidora en tres capas:

- Capa de lógica de presentación: Contendrá el código y la lógica de aplicación para devolver el código dinámico al cliente.
- Capa de lógica de negocio: Contendrá toda la lógica relativa al negocio de la aplicación. Al no estar embebida en la capa de interfaz, permitirá reducir el impacto al realizar mantenimiento, cambios o mejoras en el sistema, e incluso cambios de la interfaz de usuario.
- Capa de acceso a datos: Contendrá exclusivamente la lógica de acceso a datos del sistema.

El modo o la tecnología con la que implementaremos toda esta arquitectura en capas se detallará en el capítulo de tecnologías utilizadas.

1.4.2 *Software de cliente: Navegador web*

Para la realización del sistema utilizaremos un modelo cliente/servidor basado en servidores web, consistente en clientes sencillos que hablan con dichos servidores. En dicho modelo, los clientes y los servidores se comunican utilizando un protocolo llamado HTTP (HyperText Transfer Protocol) que define un conjunto simple de comandos en el que los parámetros son pasados como cadenas o HTTPS (la versión cifrada de HTTP basada en Secure Socket Layer) en función de los requisitos de seguridad que se necesiten.

Los clientes utilizados para comunicarnos con servidores web mediante el protocolo HTTP son los navegadores, los cuales gracias a su flexibilidad no dependerán necesariamente de una plataforma o sistema operativo determinado, sino que este quedará a elección del usuario. La función principal del navegador será descargar documentos HTML (HyperText Markup Language) y mostrarlos en pantalla al usuario.

Debido a la gran variedad de navegadores que existen en la actualidad, el objetivo es que nuestro sistema pueda utilizarse en cualquiera de ellos sin mostrar diferencia significativas.



1.4.3 Software de servidor

Debido a que se desea desarrollar un sistema que se adapte a distintos ámbitos de explotación, se intentará no obligar a la utilización de un sistema operativo en concreto. Para conseguir este objetivo, y debido a que se desea que se trate de una aplicación que se pueda utilizar mediante Internet, parte del software a utilizar vendrá dada por el lenguaje de programación que se vaya a utilizar, por lo que será lo primero que necesitemos seleccionar.

1.4.3.1 Lenguaje de programación

Descartando tecnologías de presentación antiguas, la selección del lenguaje de programación se acotará a los tres lenguajes de programación web más difundidos actualmente: PHP (Personal Home Page), .NET y Java.

En el caso de PHP, éste es un lenguaje de código abierto bastante extendido actualmente que utiliza una sintaxis similar a la de los lenguajes C o Perl. Proporciona soporte para acceso a base de datos y al ser código libre es multiplataforma además de ser aceptado por todos o casi todos los servidores web más utilizados. El problema de PHP es que al ser código interpretado en ejecución, no podemos impedir que el código sea examinado, o el coste para conseguirlo es demasiado alto para los tiempos estimados del proyecto.

El lenguaje .NET es un lenguaje que funciona en el lado de servidor y que surgió como competencia a la plataforma Java de Oracle y a PHP. Utilizando .NET podemos solventar el problema de que nuestro código sea examinado, ya que éste sólo se encuentra en los archivos del servidor por lo que los usuarios no tendrán acceso más que a la página HTML estática que se genere a partir del dicho código. El problema que encontramos en .NET es que, aunque está preparado para funcionar como multiplataforma, de momento no hay una implementación estable que permita ejecutarlo sobre cualquier sistema operativo, con lo cual nos ataríamos a tener que desplegarlo sobre una plataforma Microsoft, de ámbito privado, con el consiguiente gasto en inversión que generaría.

El último lenguaje de programación que estudiaremos y por el cual nos hemos decidido en este proyecto es Java. Este lenguaje es independiente de la plataforma y al igual que los dos lenguajes anteriores puede ejecutarse sobre cualquier tipo de hardware. En arquitecturas cliente/servidor se popularizó con la aparición de Servlets y JSP (JavaServer Pages) ya que permitían mejorar el rendimiento de tecnologías más antiguas como CGI. El hecho que ha hecho que nos decantemos por este lenguaje de programación es que aunque precisa de un contenedor dentro del cual desplegar estos componentes para poder utilizarlos en aplicaciones web dinámicas, hoy en día existen multitud de contenedores tanto libres como comerciales que funcionan en sobre cualquier plataforma, lo cual nos permite la portabilidad del código a cualquier sistema. Además dispone de multitud de frameworks de desarrollo que facilitan y aceleran la construcción, reutilización y robustez del código.



1.4.3.2 Servidor de aplicaciones

Como se ha mencionado en el apartado relativo al lenguaje de programación, al seleccionar Java para crear la aplicación web del sistema a desarrollar, será necesario disponer en la parte servidora de un contenedor en el cual podamos desplegar los servlet y páginas dinámicas que se creen.

Actualmente están disponibles una gran variedad de servidores de aplicaciones tanto de software libre como comercial. Los servidores más conocidos son Oracle WebLogic e IBM WebSphere en la parte comercial y JBoss, Apache Geronimo, Oracle GlassFish y Apache Tomcat (aunque este último no es un servidor de aplicaciones sino un contenedor servlet al no cumplir al completo la especificación J2EE).

Para la implementación del sistema se utilizará Apache Tomcat ya que, aun siendo una aplicación J2EE, en esta primera fase no se va a utilizar toda la tecnología ofrecida por esta especificación como EJB o Web Services, sino que sólo serán necesarios páginas JSP y Servlets y frameworks que utilicen estas especificaciones. Esto no quiere decir que los posibles clientes quieran desplegar el producto resultante en cualquier otro servidor, para lo cual deberá estar preparado. Además, al no tener implementada toda la especificación J2EE, el contenedor Apache Tomcat tiene unos tiempos de carga y reinicio mucho menores que el resto de aplicaciones, por lo cual es perfecto para un entorno de desarrollo.

1.4.3.3 Sistema gestor de bases de datos

La última capa software que será necesaria en la parte servidor será un sistema gestor de bases de datos que actúe como interfaz entre la base de datos, los usuarios y las aplicaciones que la utilicen.

En este caso también hay disponible una gran variedad de productos tanto comerciales (Oracle, Informix, MSAccess, ...) y como libres (MySQL, SQLite, Apache Derby, ...) que se podrán utilizar para respaldar la información del sistema a crear.

Para el entorno de desarrollo se utilizará MySQL, ya que al tener licencia GNU GPL no será necesario pagar por licencias, pero el sistema resultante debe ser capaz de ejecutarse sobre cualquier sistema gestor que quiera utilizar el usuario final, por lo que será necesario prepararlo para esta característica. Además implementa un amplio subconjunto del lenguaje SQL, está disponible para gran cantidad de plataformas, dispone de claves ajenas e indexación y se encarga de gestionar transacciones, características que serán importantes para la aplicación a desarrollar.

1.4.4 Entorno de desarrollo

Aunque el lenguaje de programación seleccionado para implementar el sistema, Java, permite su uso con un sencillo editor de texto, no es una alternativa útil a la hora de acelerar y facilitar el desarrollo. Por este motivo, se ha decidido utilizar un entorno integrado de desarrollo (Integrated development environment, IDE) que facilite dichas tareas.



Al igual que con el resto de software empleado para la realización del sistema, existen muchas alternativas a la hora de seleccionar el IDE a utilizar, tanto libres en el caso de NetBeans, JDeveloper o Eclipse, como comerciales como IntelliJ IDEA.

En este caso nos hemos decantado por Eclipse debido a su carácter gratuito, a su extensibilidad mediante plug-ins que permiten añadirle gran cantidad de funcionalidad para cubrir de las necesidades que surjan y a su fácil integración con Maven, un gestor de construcción Java que permite gestionar las dependencias de librerías, actualizar las ya existentes en el proyecto, generar versiones del producto, documentación y sobre todo, desacoplar el proyecto del IDE de desarrollo, lo cual será bastante útil si en el futuro no se quiere seguir utilizando este IDE.

1.5 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Capítulo 1:** Introducción al proyecto de fin de carrera, ambientación en la cual se va a realizar el desarrollo del sistema de información, motivación que nos ha llevado a desarrollar la aplicación, objetivos planteados que ha de cumplir el sistema para que se considere que tiene un nivel de calidad óptimo por parte de los usuarios, medios e infraestructuras de los que se va a disponer y organización del resto de la memoria.
- **Capítulo 2:** Requisitos del sistema de información. Descripción de la colección de objetivos e intenciones de uso del sistema que tienen los usuarios.
- **Capítulo 3:** Análisis del sistema y desarrollo o implementación del mismo. Diseño de la arquitectura de desarrollo.
- **Capítulo 4:** Tecnologías utilizadas y justificación de las razones que nos han impulsado a utilizar dichas tecnologías para la confección de la aplicación.
- **Capítulo 5:** Descripción del tipo de pruebas que se van a realizar, elaboración de las pruebas y tecnologías empleadas para el desarrollo de las mismas. Descripción de juego de datos que se utilizará para la realización de las pruebas.
- **Capítulo 6:** Conclusiones obtenidas tras el desarrollo del sistema, descripción del nivel de cumplimiento de los objetivos establecidos y futuros trabajos que ayudarán a mejorar el mismo.
- **Anexos:** En los anexos emplazaremos el manual de usuario, el manual del administrador de la aplicación y el presupuesto y la planificación elaborados para la implementación de la misma, desde el análisis hasta la implantación y explotación por parte del usuario.



2 Requisitos

2.1 – Restricciones

2.3 – Requisitos funcionales

2.2 – Suposiciones

2.4 – Requisitos no funcionales

Este apartado desarrolla la Especificación de Requisitos de Software (ERS) para el sistema de Gestión de Espacios y Eventos Culturales. El objeto de esta especificación de requisitos será definir de manera clara y concisa todas las funcionalidades y restricciones del sistema a construir, ya que este apartado irá dirigido tanto al equipo de desarrollo como a los usuarios con los que nos entrevistaremos para su realización.

Para el proceso de extracción de requisitos hemos utilizado dos técnicas distintas. Por un lado, hemos utilizado la técnica de observación, ya que debido a que hemos trabajado con personas encargadas del proceso de programación cultural, hemos podido recoger datos de primera mano. También hemos utilizado la técnica de entrevistas con las personas con las que colaborábamos, lo cual nos ha permitido explorar el sujeto del proyecto y obtener datos cualitativos del mismo.

2.1 Restricciones

A continuación enumeraremos las restricciones del sistema no proveniente de los requisitos, que asumiremos a la hora de construcción del sistema:

- Se diseñará el sistema siguiendo un modelo de programación Web.
- Las comunicaciones del sistema estarán basadas en protocolos estándar de Internet.
- Utilizar UML (Unified Modeling Language) como metodología orientada a objetos que proporcione un modelado común para todas las iteraciones del desarrollo y permita que cualquier analista o desarrollador sea capaz de comprender el sistema independientemente del lenguaje de implementación utilizado.
- En un principio el sistema se construirá como software libre, ya que su primera implantación posiblemente sea en la administración pública y sus desarrollos deben tener algunas de las licencias que cumplan esta restricción.



2.2 Suposiciones

Asumiremos que los requisitos que describiremos en este apartado serán estables una vez comenzado el desarrollo y que cualquier cambio en los mismos deberá plantearse para una siguiente iteración del sistema.

2.3 Requisitos funcionales

Los Requisitos Funcionales son los sujetos fundamentales o esenciales que constituyen la médula del producto. Describen lo que el producto tiene que hacer o qué acciones de procesamiento debe tomar.

Todos los requisitos aquí expuestos serán ESENCIALES, es decir, no sería aceptable un sistema que no satisfaga alguno de los requisitos aquí presentados. Estos requisitos se especificarán teniendo en cuenta, entre otros, el criterio de testeabilidad, es decir, que debería ser fácilmente demostrable si es satisfecho o no por el sistema.

En el sistema a desarrollar se han identificado los siguientes requisitos funcionales:

2.3.1 *RFUN01: Autenticar usuarios*

El sistema autenticará a los usuarios para que puedan acceder al mismo en función de un perfil y de un grupo de trabajo, pudiendo los mismos tener varios perfiles en un mismo o distintos grupos.

2.3.2 *RFUN02: Creación de grupos de trabajo*

El sistema permitirá al usuario superadministrador la creación de grupos de trabajo a los cuales asociar usuarios y para los cuales se podrán gestionar espacios y eventos.

2.3.3 *RFUN03: Listado de grupos de trabajo*

El usuario superadministrador dispondrá de un listado de los grupos de trabajo existentes.

2.3.4 *RFUN04: Edición de grupos de trabajo*

El sistema permitirá que el usuario superadministrador edite o modifique grupos de trabajo existentes.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	--	--

2.3.5 RFUN05: Borrado de grupos de trabajo

El sistema permitirá que el usuario superadministrador elimine grupos de trabajo.

2.3.6 RFUN06: Gestión de empleados

El sistema permitirá gestionar los empleados del organismo o empresa introducidos en la aplicación. En un principio sólo interesa una gestión superficial de los mismos, pero a la hora de diseñar el modelo de datos, deberá pensarse en una futura ampliación de la información almacenada para los empleados.

2.3.7 RFUN07: Listado de empleados

El sistema permitirá mostrar un listado de los empleados, con o sin usuario.

2.3.8 RFUN08: Edición de empleados

El sistema permitirá que el usuario superadministrador modifique los datos de los empleados introducidos en la aplicación.

2.3.9 RFUN09: Borrado de empleados

El sistema permitirá que el usuario superadministrador elimine empleados del sistema. Si se elimina un empleado se eliminará también su usuario y los roles que tenga en los distintos grupos.

2.3.10 RFUN10: Creación de usuarios

El sistema permitirá que el usuario superadministrador cree cualquier perfil de usuario en la aplicación.

2.3.11 RFUN11: Borrado de usuarios

El sistema permitirá que el usuario superadministrador elimine usuarios del mismo.

2.3.12 RFUN12: Creación de perfil de usuario en grupo

El sistema permitirá que el usuario superadministrador y el administrador añadan roles de usuario en un determinado grupo.



2.3.13 RFUN13: Borrado de perfil de usuario en grupo

El sistema permitirá que el usuario superadministrador y el administrador eliminen roles de usuario en un determinado grupo.

2.3.14 RFUN14: Creación de contactos

El sistema permitirá que cualquier usuario cree fichas de contactos que serán compartidas entre todos los usuarios.

2.3.15 RFUN15: Listado y búsqueda de contactos

El sistema dispondrá de un listado de contactos que podrá ser consultado por todos los usuarios y sobre el cual se podrán realizar búsquedas.

2.3.16 RFUN16: Edición de contactos

El sistema permitirá que cualquier usuario modifique las fichas de contactos.

2.3.17 RFUN17: Borrado de contactos

El sistema permitirá la eliminación de contactos únicamente al usuario superadministrador, ya que se trata de información compartida por todos los grupos.

2.3.18 RFUN18: Creación de espacios

El sistema permitirá que el usuario superadministrador y el administrador de grupo creen nuevos espacios escénicos para un grupo de trabajo.

2.3.19 RFUN19: Listado de espacios

El sistema dispondrá de un listado de espacios escénicos por grupo de pertenencia, excepto para el usuario superadministrador que podrá consultar todos los espacios disponibles en todos los grupos.

2.3.20 RFUN20: Edición de espacios

El sistema permitirá que el usuario superadministrador y el administrador de un grupo modifiquen los

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

espacios escénicos pertenecientes a un grupo.

2.3.21 RFUN21: Borrado de espacios

El sistema permitirá que el usuario superadministrador y el administrador de grupo eliminen espacios escénicos de cualquier grupo de trabajo o del grupo de trabajo al que pertenece, respectivamente.

2.3.22 RFUN22: Creación de eventos

El sistema permitirá crear eventos culturales a los usuarios con el perfil de editor en un determinado grupo o superadministrador. Estos eventos se crearán en estado pendiente de autorización hasta que sean validados por un usuario con perfil autorizador.

2.3.23 RFUN23: Listado de eventos

El sistema dispondrá de un listado de eventos culturales por grupo de pertenencia al cual podrán acceder los usuarios administrador, editor y autorizador de grupo y el usuario superadministrador de la aplicación. Para los usuarios de un grupo sólo se mostrarán los eventos del grupo y para el usuario superadministrador se mostrarán todos los eventos de todos los grupos.

2.3.24 RFUN24: Planificación de eventos

El sistema permitirá que todos los usuarios consulten los eventos planificados en la aplicación. Además los usuarios editores de grupo deberán poder planificar eventos basándose en la agenda que muestra el listado de eventos planificados.

2.3.25 RFUN25: Edición de eventos

El sistema permitirá que los usuarios con el perfil de editor en un determinado grupo modifiquen los eventos del mismo siempre que el mismo no se encuentre ya en estado aceptado. Los usuarios con perfil autorizador podrán modificar el evento pero únicamente para autorizarlo o cancelarlo. Al modificar un evento, se deberá proceder nuevamente a la revisión del mismo para su autorización o rechazo.

2.3.26 RFUN26: Autorización o rechazo de eventos

El sistema permitirá que los usuarios con el perfil de autorizador en un determinado grupo autoricen o rechacen los eventos de dicho grupo.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

Los eventos que no hayan sido autorizados antes de 15 días de su fecha de ejecución pasarán automáticamente a estado rechazado.

2.3.27 RFUN27: Borrado de eventos

El sistema eliminará los eventos rechazados cuya fecha de programación sea superior a la fecha actual.

2.3.28 RFUN28: Actualización del histórico de eventos

El sistema moverá automáticamente los eventos ejecutados cuya fecha sea anterior a un año, contando desde la fecha actual, a un histórico de eventos.

2.3.29 RFUN29: Histórico de eventos

El sistema proporcionará un listado del histórico de eventos ya ejecutados. El contenido del listado dependerá del grupo al cual pertenezca el usuario, mostrándose exclusivamente los eventos de su grupo, excepto en el caso del superadministrador, que se mostrarán todos los eventos.

2.3.30 RFUN30: Paginación de listados

Todos los listados de la aplicación deberán estar paginados en grupos de 10 elementos permitiendo al usuario navegar entre las distintas páginas de datos.

2.3.31 RFUN31: Filtros de búsqueda

Todos los listados de la aplicación deberán disponer de un sistema de filtros de búsqueda, que permitan filtrar el contenido del listado en función de los principales campos de cada entidad mostrada.

2.4 Requisitos no funcionales

Los Requisitos NO Funcionales son las propiedades que las funciones deben tener, tales como facilidad de uso y seguridad.

En el sistema a desarrollar se han identificado los siguientes requisitos no funcionales:

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

2.4.1 Requisitos de datos

Son los requisitos que indicarán los datos que habrá que almacenar en el sistema para llevar a dato la gestión de espacios y eventos culturales.

2.4.1.1 RDAT01: Datos sobre usuarios

El sistema almacenará datos sobre los usuarios registrados en el sistema. Entre los datos que almacenaremos en el sistema se encuentran los datos del perfil de usuario y el grupo al que pertenece.

2.4.1.2 RDAT02: Datos de autenticación de usuarios

El sistema almacenará los nombres y contraseñas cifradas de todos los usuarios.

2.4.1.3 RDAT03: Datos sobre los grupos

El sistema almacenará los datos de los distintos grupos que podrán utilizar el sistema para gestionar su programación de espacios y eventos culturales.

2.4.1.4 RDAT04: Datos sobre actividad de usuario

El sistema deberá registrar y almacenar una traza de las operaciones realizada por cada uno de sus usuarios.

2.4.1.5 RDAT05: Datos sobre contactos y organizaciones

El sistema almacenará datos sobre contactos y organizaciones a las que pertenecen. Entre otros datos se almacenará su nombre, el teléfono de contacto, etc.

2.4.1.6 RDAT06: Datos sobre espacios

El sistema almacenará el nombre, ubicación y grupo al que pertenece, y opcionalmente los aspectos técnicos del espacio que se desee que sea gestionado por un grupo y al que se puedan asignar eventos culturales.

2.4.1.7 RDAT07: Datos sobre eventos

El sistema deberá almacenar datos sobre los eventos programados por los distintos grupos sobre los espacios gestionados por ellos. Además deberá almacenar el estado en que se encuentre dicho evento.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

2.4.2 Requisitos de seguridad

Especifican quién tiene acceso autorizado al producto, tanto a funcionalidad como a los datos, bajo qué circunstancias se garantiza el acceso, y a qué partes del producto se permite el acceso.

2.4.2.1 RSEG01: Privacidad de usuarios y de datos sensibles almacenados

En la implementación del sistema se ha de tener especial cuidado con la privacidad y la seguridad de los datos de los usuarios y otra información sensible que se pueda almacenar, teniendo en cuenta la Ley 15/1999 de protección de datos de carácter personal. Por tanto los datos sensibles que se almacenen en la misma deberán ir convenientemente cifrados y dejar registro del acceso a los mismos para posteriores auditorías.

2.4.2.2 RSEG02: Existencia de un usuario base o superadministrador

El sistema deberá tener como mínimo, tanto en su primera ejecución como durante toda la vida del mismo, un usuario base que denominaremos “superadministrador”, que será el encargado de realizar la configuración inicial del sistema entre cuyas tareas se encontrará la gestión de grupos y de usuarios de sistema.

2.4.2.3 RSEG03: Perfiles de usuario

La aplicación sólo admitirá tres perfiles de usuario distintos, que son: administrador de grupo, editor de grupo y autorizador de grupo.

2.4.2.4 RSEG04: Acceso mediante perfiles y grupos

Sólo podrán acceder al sistema los usuarios que dispongan de un perfil adecuado en uno o varios grupos, a excepción del superadministrador que no necesitará estar asignado a un determinado grupo.

2.4.2.5 RSEG05: Sesiones por usuario

El sistema sólo permitirá una conexión simultánea por cada usuario dado de alta en el mismo para solventar el problema de suplantación de sesión.

2.4.3 Requisitos de interfaces de usuario

Especifican requisitos de apariencia y requisitos que harán que el producto sea utilizable y configurable según las preferencias del usuario.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	--	--

2.4.3.1 RIUS01: Internacionalización

El sistema deberá permitir que todos los textos de sus interfaces puedan adaptarse a varios idiomas y que el usuario pueda cambiar fácilmente entre ellos.

2.4.3.2 RIUS02: Retro-alimentación al usuario

El sistema debe proporcionar retro-alimentación al usuario en el caso de entrada incorrecta de datos, operaciones en progreso, etc.

2.4.3.3 RIUS03: Información de ayuda a los usuarios

El sistema proporcionará información de ayuda al usuario para cada interfaz con la que este interactúe.

2.4.3.4 RIUS04: Menú de navegación

El sistema siempre tendrá visible un menú de navegación que mostrará u ocultará distintas partes u opciones en función del perfil del usuario autenticado en el mismo.

2.4.4 Requisitos de interfaces externos

Especifican requisitos necesarios para que el sistema pueda interactuar con aplicaciones externas y/o con dispositivos que el producto necesite para funcionar correctamente.

2.4.4.1 RIEX01: Compatibilidad en navegadores

El sistema deberá funcionar correctamente sobre los cuatro navegadores más populares existentes en el momento de realización del mismo.

2.4.5 Requisitos de interfaces de comunicación

2.4.5.1 RCOM01: Protocolos de comunicación.

Los protocolos de comunicación del sistema (entre las distintas capas o con el cliente del usuario) serán los protocolos estándares de Internet.

2.5 Requisitos de rendimiento

Especifican requisitos relativos a la habilidad del sistema para operar a una velocidad adecuada en el

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

ambiente previsto, así como a los volúmenes y cantidad de datos que el producto será capaz de manejar.

2.5.1.1 RREN01: Tiempos de respuesta en peticiones.

Los tiempos de respuesta para las peticiones de los clientes deben ser bajos, suponiendo que tenemos una red de comunicaciones óptima.

2.5.1.2 RREN02: Concurrencia de usuarios.

El sistema debe soportar el acceso concurrente de usuarios a la aplicación.

2.5.1.3 RREN03: Paginación de datos.

El sistema debe soportar la paginación de datos sin tener que almacenarlos en memoria, ya que hay que prever el crecimiento de los mismos en función del uso.

2.6 Requisitos de desarrollo

Especifican restricciones o necesidades relativas a la implementación del sistema.

2.6.1.1 RDES01: Arquitectura de sistema.

El sistema debe establecer una arquitectura de desarrollo que permita la modularización y facilite las futuras ampliaciones del mismo.

2.6.1.2 RDES02: Ciclo de vida.

El ciclo de vida elegido para desarrollar el sistema será el de prototipo evolutivo, de manera que se puedan incorporar fácilmente cambios y nuevas funcionalidades.

2.7 Requisitos tecnológicos

No se ha definido ningún requisito tecnológico.

2.8 Atributos del sistema

Atributos que ha de tener el sistema a desarrollar para su aceptación final.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	--	--

2.8.1 Sistema portable.

El sistema debe ser portable y por tanto tanto el mismo como cualquier componente software de que dependa debe poder instalarse en diversas plataformas.

2.8.2 Evolutivo.

El sistema debe estar preparado para añadir evoluciones o ampliaciones al mismo, por lo que debe documentarse el código y ha de ser modular.



3 Análisis y diseño

3.1 – Análisis de casos de uso**3.3 – Diagramas de clases (estructurales)****3.2 – Modelo de datos****3.4 – Diagramas de objetos y de secuencia**

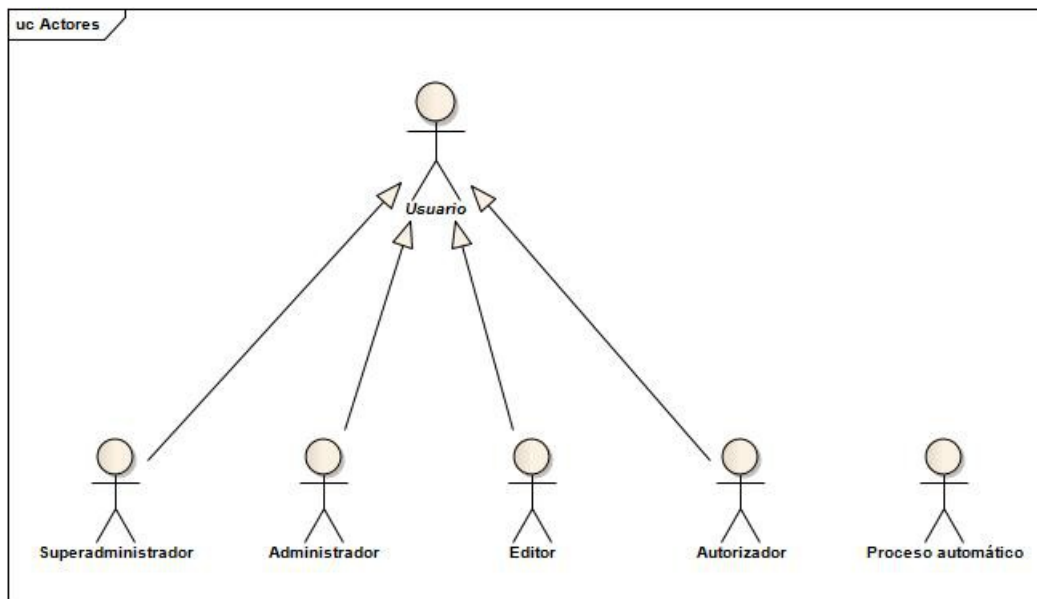
Una vez establecido el objetivo principal del proyecto y los objetivos a nivel de sistema que se deben alcanzar, en este apartado se procederá a realizar el análisis y diseño del sistema de información. En este apartado, se incluirá el análisis de casos de uso de los requisitos de sistema lo cual facilitará la comunicación con el cliente para identificar las interacciones y los límites del sistema, y a continuación, se recogerán varios apartados relativos al diseño de la aplicación en los que se intentará trasladar la información obtenida en el análisis de requisitos a un enfoque centrado en el diseño de una solución que satisfaga los mismos.

3.1 Análisis de casos de uso

A partir de los requisitos de sistema identificados en la sección anterior, se ha obtenido el modelo de casos de uso que se desarrolla en los siguientes apartados. Este modelo es un catálogo de la funcionalidad del sistema descrita utilizando casos de uso UML. Cada caso de uso representa una interacción única y repetible que experimentará un usuario o “actor” cuando utilice el sistema.

Los casos de uso son un artefacto de la especificación UML que nos proporcionarán visibilidad de las operaciones que se requiere que realice el sistema, desde el punto de vista de los actores. Los casos de uso se han extraído de los requisitos definidos por el cliente, lo cual proporciona una herramienta gráfica de comunicación con el mismo. Además, un caso de uso incluye uno o más “escenarios” los cuales describen la interacción entre el actor y el sistema, y documenta los resultados y excepciones que ocurran desde el punto de vista de los usuarios.

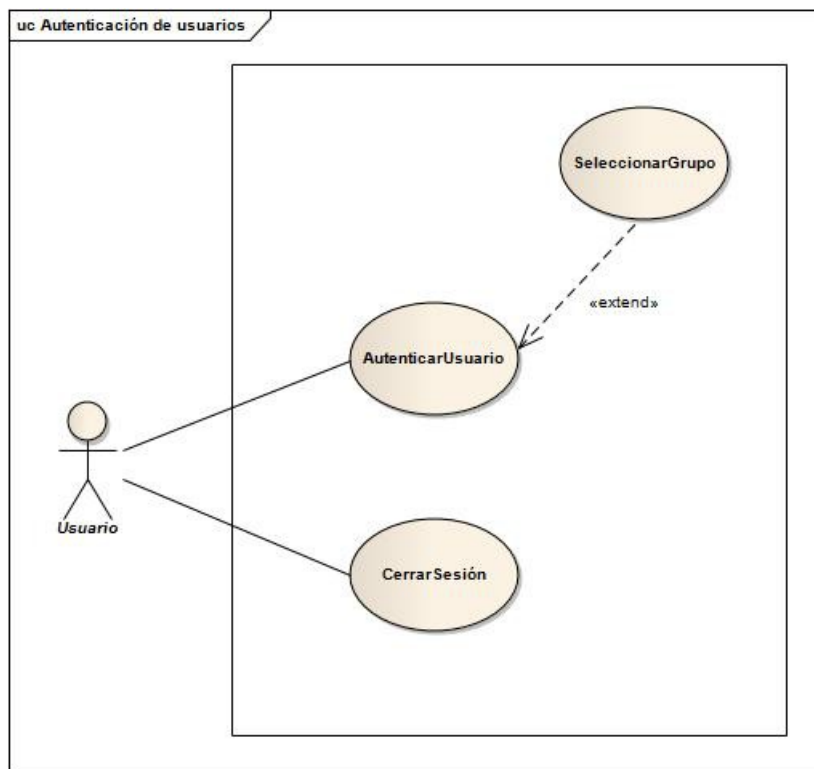
En el sistema a desarrollar se ha diferenciado entre los siguientes tipos de actores:



- Superadministrador: El actor superadministrador es el que configura la aplicación pudiendo crear nuevos grupos, usuarios de los grupos existentes, borrado de datos comunes a todos los grupos, etc.
- Administrador: El actor administrador será el encargado de gestionar un determinado grupo dado de alta en la aplicación, realizando todas las operaciones necesarias sobre los usuarios y los espacios. También tendrá acceso a las operaciones comunes del sistema como la gestión de contactos.
- Editor: El actor editor es el que utiliza la aplicación como herramienta de planificación para los distintos eventos culturales a programar para un determinado grupo.
- Autorizador: El actor autorizador se encargará de autorizar o rechazar, antes de la fecha límite establecida por los requisitos, los eventos programados en un determinado grupo.
- Proceso automático: El proceso automático será un usuario especial que no será una persona física, y que se encargará de comprobar el sistema de forma recurrente en ciclos programados, para interactuar con los eventos y realizar sobre ellos distintas tareas automáticas.

Por tanto, al analizar el texto en busca de los verbos que denoten uso o interacción con el sistema, se obtiene la lista de casos de uso que se expone a continuación, agrupados por bloque de funcionalidad.

3.1.1 Autenticación de usuarios



3.1.1.1 Autenticar usuario

Actores: Usuario (Superadministrador, Administrador, Editor, Autorizador)

Tipo: Principal

Descripción: Este caso de uso se corresponde con el proceso de autenticación para poder acceder a la aplicación.

Requisitos: Acceder a la página principal de la aplicación.

Escenario de éxito:



Acciones de los actores	Respuesta del sistema
1. El usuario accede a la página principal de la aplicación	
	2. El sistema le muestra el formulario de autenticación de usuario con los campos de usuario y contraseña.
3. El usuario introduce los datos de autenticación.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema autentica al usuario en la aplicación y carga sus datos en la sesión de la misma.
	6. El sistema le muestra al usuario la página principal de la aplicación en función de su perfil o perfiles de usuario.

Extensiones:**4a. Selección del grupo de usuario.**

Acciones de los actores	Respuesta del sistema
1. El usuario ha introducido los datos correctos de autenticación.	
	2. El sistema comprueba los grupos a los que pertenece el usuario.
	3. El sistema muestra el formulario de selección de grupo.
4. El usuario selecciona el grupo en el cual autenticarse, en caso de estar asignado en más de un grupo.	

3.1.1.2 Cerrar sesión

Actores: Usuario (Superadministrador, Administrador, Editor, Autorizador)

Tipo: Principal

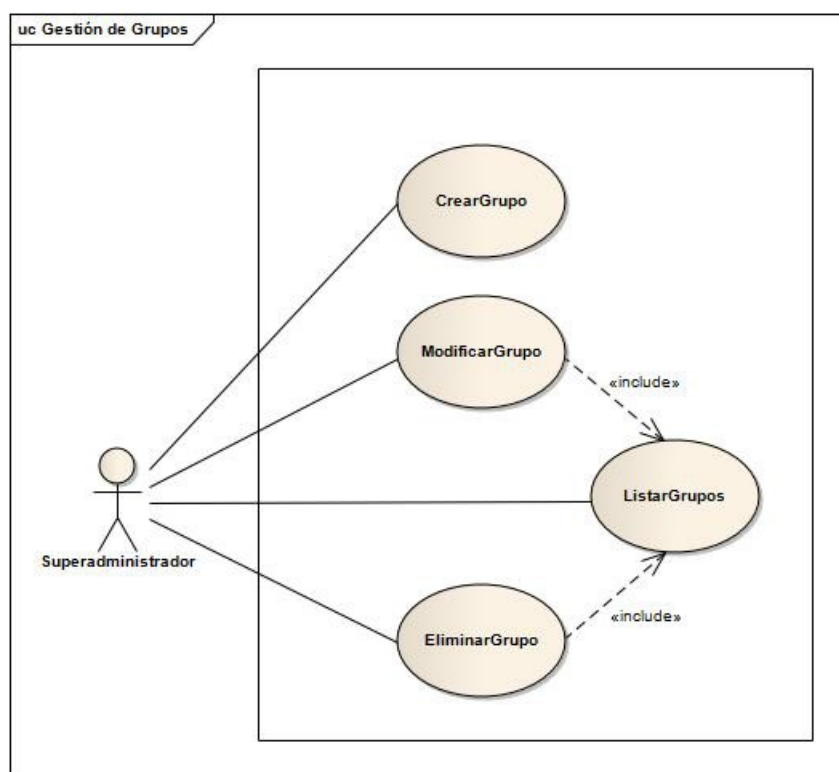
Descripción: Este caso de uso se corresponde con la acción de cierre de sesión de usuario en la aplicación una vez que ha terminado de trabajar con la misma.

Requisitos: Estar autenticado en el sistema.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el cierre de sesión.	
	2. El sistema elimina los datos de autenticación de usuario de la sesión de la aplicación.
	3. El sistema muestra al usuario el formulario de autenticación para acceder a la aplicación.

3.1.2 Gestión de grupos



3.1.2.1 Crear grupo

Actores: Superadministrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con el proceso de crear un nuevo grupo de usuarios, espacios y eventos.

Requisitos: El usuario debe estar autenticado en la aplicación y tener el rol de superadministrador.

Escenario de éxito:



Acciones de los actores	Respuesta del sistema
1. El usuario accede a la operación de creación de grupo.	
	2. El sistema muestra el formulario de introducción de datos de grupo.
3. El usuario introduce los datos del nuevo grupo y los acepta.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema registra el nuevo grupo.
	6. El sistema muestra al usuario el mensaje de resultado de la aplicación.

3.1.2.2 Listar grupos

Actores: Superadministrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de mostrar el listado de grupos de usuarios, espacios y eventos.

Requisitos: El usuario debe estar autenticado en la aplicación y tener el rol de superadministrador.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la opción de mostrar el listado de grupos.	
	2. El sistema consulta los grupos dados de alta en la aplicación.
	3. El sistema muestra al usuario el listado de grupos.

Escenario alternativo:

3a. No se han dado de alta grupos en el sistema.

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de información de no existencia de grupos.



3.1.2.3 *Modificar grupo*

Actores: Superadministrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de modificar un grupo de usuarios, espacios y eventos.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el rol de superadministrador y acceder a la opción de listado de grupos.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el grupo a modificar del listado de grupos.	
	2. El sistema muestra al usuario el formulario de edición de datos de grupo.
3. El usuario modifica los datos del grupo que considere pertinentes.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema actualiza los datos del grupo.
	6. El sistema muestra al usuario el listado de grupos actualizado y el mensaje de resultado de la operación.

3.1.2.4 *Eliminar grupo*

Actores: Superadministrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de eliminar un grupo de usuarios, espacios y eventos.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el rol de superadministrador y acceder a la opción de listado de grupos.

Escenario de éxito:



Gestión de espacios y eventos culturales (GEEC)

Proyecto de fin de carrera

Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el grupo a eliminar del listado de grupos.	
	2. El sistema muestra al usuario el formulario de confirmación de eliminación con los datos del grupo eliminar.
3. El usuario pulsa el botón de confirmación de borrado.	
	4. El sistema comprueba que el grupo no tenga usuarios y/o espacios asignados.
	5. El sistema elimina el grupo.
	6. El sistema muestra al usuario el listado de grupos actualizado y el mensaje de resultado de la operación.

Escenario alternativo:

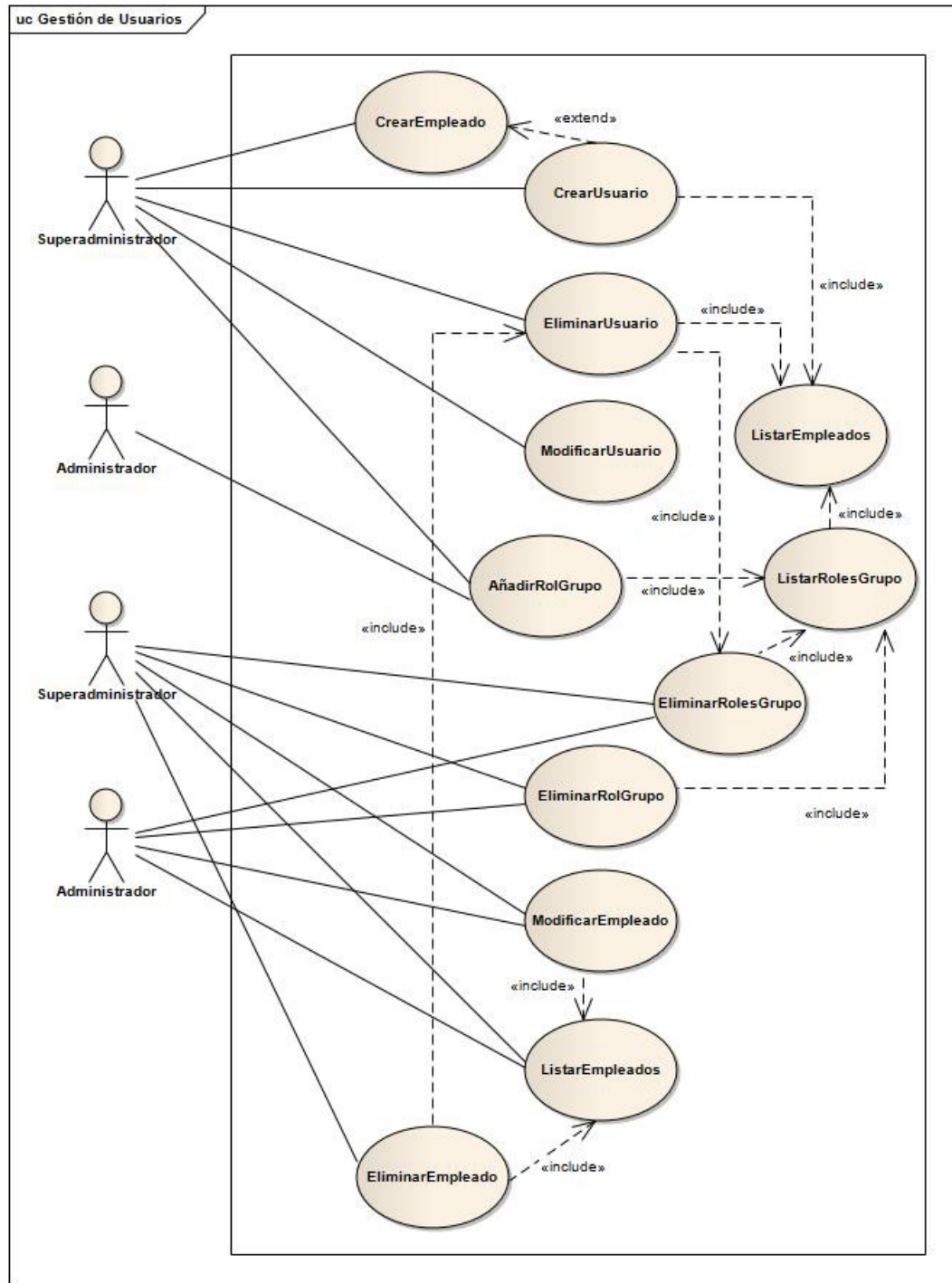
4a. El grupo tiene usuarios.

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de error informando de que el grupo tiene usuarios asignados y el listado inicial de grupos.

4b. El grupo tiene espacios asignados.

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de error informando de que el grupo tiene espacios asignados y el listado inicial de grupos.

3.1.3 Gestión de usuarios



	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

3.1.3.1 Crear empleado

Actores: Superadministrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con el proceso de dar de alta un nuevo empleado en la aplicación.

Requisitos: El usuario debe estar autenticado en la aplicación y tener el rol de superadministrador.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la operación de creación de empleado.	
	2. El sistema muestra el formulario de introducción de datos de empleado.
3. El usuario introduce los datos del nuevo empleado y los acepta.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema registra el nuevo empleado.
	6. El sistema muestra al usuario el mensaje de resultado de la aplicación.

Extensiones:

5a. Creación de empleado con usuario.

Acciones de los actores	Respuesta del sistema
1. El usuario selecciona la opción del formulario de crear el usuario del empleado.	
	2. El sistema muestra el formulario de creación de usuario de empleado.
3. El usuario introduce los datos del nuevo usuario de grupo.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema registra el nuevo usuario asociado al empleado.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

3.1.3.2 Listar empleados

Actores: Superadministrador, Administrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de mostrar el listado de empleados y usuarios.

Requisitos: El usuario debe estar autenticado en la aplicación y tener el rol de superadministrador o administrador.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la opción de mostrar el listado de empleados.	
	2. El sistema consulta los empleados dados de alta en la aplicación.
	3. El sistema muestra al usuario el listado de empleados de su grupo o en caso del superadministrador el listado completo de empleados.

Escenario alternativo:

3a. No se han dado de alta empleados en el grupo o en el sistema.

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de información de no existencia de empleados.

3b. Filtrado del listado de empleados mostrado.

Acciones de los actores	Respuesta del sistema
1. El usuario introduce los valores de los filtros de búsqueda que desea aplicar sobre el listado de empleados y pulsa el botón buscar.	
	2. El sistema actualiza el listado de empleados aplicando los filtros de búsqueda.

3c. Cancelación de filtros de búsqueda de empleados.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

Acciones de los actores	Respuesta del sistema
1. El usuario pulsa el botón de limpiar filtros de búsqueda.	
	2. El sistema restaura el listado de empleados.

3.1.3.3 Crear usuario

Actores: Superadministrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con el proceso de crear un nuevo usuario en un grupo.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el rol de superadministrador y acceder al listado de empleados.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la operación de creación de usuario.	
	2. El sistema muestra el formulario de introducción de datos de usuario.
3. El usuario introduce los datos del nuevo usuario de grupo y los acepta.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema registra el nuevo usuario.
	6. El sistema muestra al usuario el mensaje de resultado de la aplicación.

3.1.3.4 Modificar usuario

Actores: Superadministrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de modificar un usuario de grupo.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el rol de superadministrador y acceder a la opción de listado de empleados.

Escenario de éxito:



Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el usuario a modificar del listado de empleados.	
	2. El sistema muestra al usuario el formulario de edición de datos de usuario.
3. El usuario modifica los datos del usuario que considere pertinentes.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema actualiza los datos del usuario.
	6. El sistema muestra al usuario el listado de empleados actualizado y el mensaje de resultado de la operación.

3.1.3.5 Eliminar usuario

Actores: Superadministrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de eliminar un usuario de grupo.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el rol de superadministrador y acceder a la opción de listado de empleados.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el usuario a eliminar del listado de empleados.	
	2. El sistema muestra al usuario el formulario de confirmación de eliminación con los datos del usuario a eliminar.
3. El usuario pulsa el botón de confirmación de borrado.	
	4. El sistema elimina el usuario.
	5. El sistema elimina los perfiles de grupo del usuario.
	6. El sistema muestra al usuario el listado de empleados actualizado y el mensaje de resultado de la operación.



3.1.3.6 Listar roles de usuario en grupo

Actores: Superadministrador, Administrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de listar los roles que tiene un usuario en un determinado grupo, o grupos en caso del superadministrador.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el rol de superadministrador o administrador y acceder a la opción de listado de empleados.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el usuario para el cual listar sus perfiles en un grupo o grupos.	
	2. El sistema muestra al usuario el listado de listado de roles de usuario en grupo o grupos.

Escenario alternativo:

2a. No se han dado de alta roles para el usuario en ningún grupo.

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de información de no existencia de perfiles para el usuario en ningún grupo.

3.1.3.7 Añadir rol de usuario en grupo

Actores: Superadministrador, Administrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de creación de un perfil para un usuario en un determinado grupo.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el rol de superadministrador o administrador y acceder a la opción de listado de empleados.

Escenario de éxito:



Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el usuario para el cual añadir el perfil en el grupo.	
	2. El sistema muestra al usuario el listado de perfiles de usuario y grupos.
3. El usuario selecciona la opción de añadir perfil en grupo.	
	4. El sistema muestra el formulario de creación de perfil en grupo.
5. El usuario selecciona el perfil a añadir en el grupo que administra o en el caso del superadministrador, selecciona también el grupo.	
	6. El sistema comprueba que se han introducido los datos obligatorios.
	7. El sistema añade el perfil del usuario en el grupo.
	8. El sistema muestra al usuario el listado de gestión de perfiles y grupos actualizado y el mensaje de resultado de la operación.

3.1.3.8 Eliminar rol de usuario de grupo

Actores: Superadministrador, Administrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de eliminar un determinado perfil de un usuario en un grupo.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el rol de superadministrador o administrador y acceder a la opción de listado de empleados.

Escenario de éxito:

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el usuario para el cual eliminar el perfil del grupo.	
	2. El sistema muestra al usuario el listado de perfiles de usuario y grupos.
3. El usuario selecciona la opción de eliminar perfil de grupo. En el caso del superadministrador seleccionará también el grupo.	
	4. El sistema muestra el formulario de confirmación de eliminación de perfil de grupo.
5. El usuario selecciona el perfil de grupo a eliminar y acepta la operación.	
	7. El sistema elimina el perfil del usuario en el grupo.
	8. El sistema muestra al usuario el listado de gestión de perfiles y grupos actualizado y el mensaje de resultado de la operación.

3.1.3.9 *Eliminar todos los roles de usuario de grupo*

Actores: Superadministrador, Administrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de eliminar un usuario de grupo.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el rol de superadministrador o administrador y acceder a la opción de listado de empleados.

Escenario de éxito:



Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el usuario para el cual eliminar todos los perfiles del grupo.	
	2. El sistema muestra al usuario el listado de perfiles de usuario y grupos.
3. El usuario selecciona la opción de eliminar todos los perfiles del grupo. En el caso del superadministrador seleccionará también el grupo.	
	4. El sistema muestra el formulario de confirmación de eliminación de perfiles de grupo.
5. El usuario acepta la operación.	
	7. El sistema elimina todos los perfiles del usuario en el grupo seleccionado.
	8. El sistema muestra al usuario el listado de gestión de perfiles y grupos actualizado y el mensaje de resultado de la operación.

3.1.3.10 *Modificar empleado*

Actores: Superadministrador, Administrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de modificar los datos de un empleado dado de alta en la aplicación.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el rol de superadministrador o de administrador y acceder a la opción de listado de empleados.

Escenario de éxito:



Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el empleado a modificar del listado de empleados.	
	2. El sistema muestra al usuario el formulario de edición de datos de empleado.
3. El usuario modifica los datos del empleado que considere pertinentes.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema actualiza los datos del empleado.
	6. El sistema muestra al usuario el listado de empleados actualizado y el mensaje de resultado de la operación.

3.1.3.11 Eliminar empleado

Actores: Superadministrador

Tipo: Principal

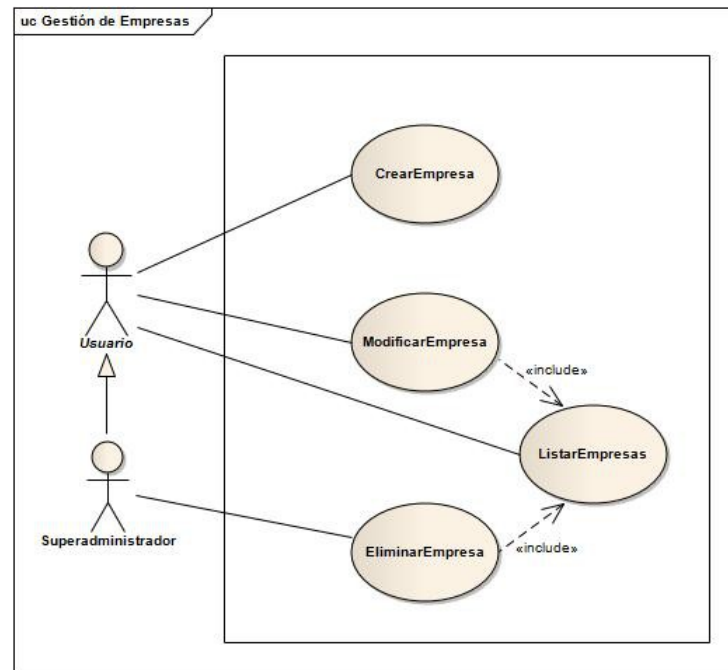
Descripción: Este caso de uso se corresponde con la operación de eliminar un empleado.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el rol de superadministrador y acceder a la opción de listado de empleados.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el empleado a eliminar del listado de empleados.	
	2. El sistema muestra al usuario el formulario de confirmación de eliminación con los datos del empleado a eliminar.
3. El usuario pulsa el botón de confirmación de borrado.	
	4. El sistema elimina los perfiles del usuario en todos los grupos.
	5. El sistema el usuario asociado al empleado.
	6. El sistema elimina el empleado.
	7. El sistema muestra al usuario el listado de empleados actualizado y el mensaje de resultado de la operación.

3.1.4 Gestión de empresas



3.1.4.1 Crear empresa

Actores: Usuario (Superadministrador, Administrador, Editor, Autorizador)

Tipo: Principal

Descripción: Este caso de uso se corresponde con el proceso de crear una nueva empresa.

Requisitos: El usuario debe estar autenticado en la aplicación.

Escenario de éxito:



Gestión de espacios y eventos culturales (GEEC)

Proyecto de fin de carrera

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la operación de creación de empresa.	
	2. El sistema muestra el formulario de introducción de datos de la empresa.
3. El usuario introduce los datos de la nueva empresa y los acepta.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema registra la nueva empresa.
	6. El sistema muestra al usuario el mensaje de resultado de la aplicación.

3.1.4.2 Listar empresas

Actores: Usuario (Superadministrador, Administrador, Editor, Autorizador)

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de mostrar el listado de empresas.

Requisitos: El usuario debe estar autenticado en la aplicación.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la opción de mostrar el listado de empresas.	
	2. El sistema consulta las empresas dadas de alta en la aplicación.
	3. El sistema muestra al usuario el listado de empresas. El listado de empresas debe mostrar principalmente la información de contacto de la empresa.

Escenario alternativo:

3a. No se han dado de alta empresas en el sistema.

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de información de no existencia de empresas.

3b. Filtrado del listado de empresas mostrado.

Acciones de los actores	Respuesta del sistema
1. El usuario introduce los valores de los filtros de búsqueda que desea aplicar sobre el listado de empresas y pulsa el botón buscar.	
	2. El sistema actualiza el listado de empresas aplicando los filtros de búsqueda.

3c. Cancelación de filtros de búsqueda de empresas.

Acciones de los actores	Respuesta del sistema
1. El usuario pulsa el botón de limpiar filtros de búsqueda.	
	2. El sistema restaura el listado de empresas.

3.1.4.3 *Modificar empresa*

Actores: Usuario (Superadministrador, Administrador, Editor, Autorizador)

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de modificar una empresa.

Requisitos: El usuario debe estar autenticado en la aplicación y acceder a la opción de listado de empresas.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario selecciona la empresa a modificar del listado de empresa.	
	2. El sistema muestra al usuario el formulario de edición de datos de empresa.
3. El usuario modifica los datos de la empresa que considere pertinentes.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema actualiza los datos de la empresa.
	6. El sistema muestra al usuario el listado de empresas actualizada y el mensaje de resultado de la operación.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

3.1.4.4 Eliminar empresa

Actores: Superadministrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de eliminar una empresa.

Requisitos: El usuario debe estar autenticado en la aplicación y acceder a la opción de listado de empresas. El usuario debe tener rol de superadministrador de la aplicación.

Escenario de éxito:

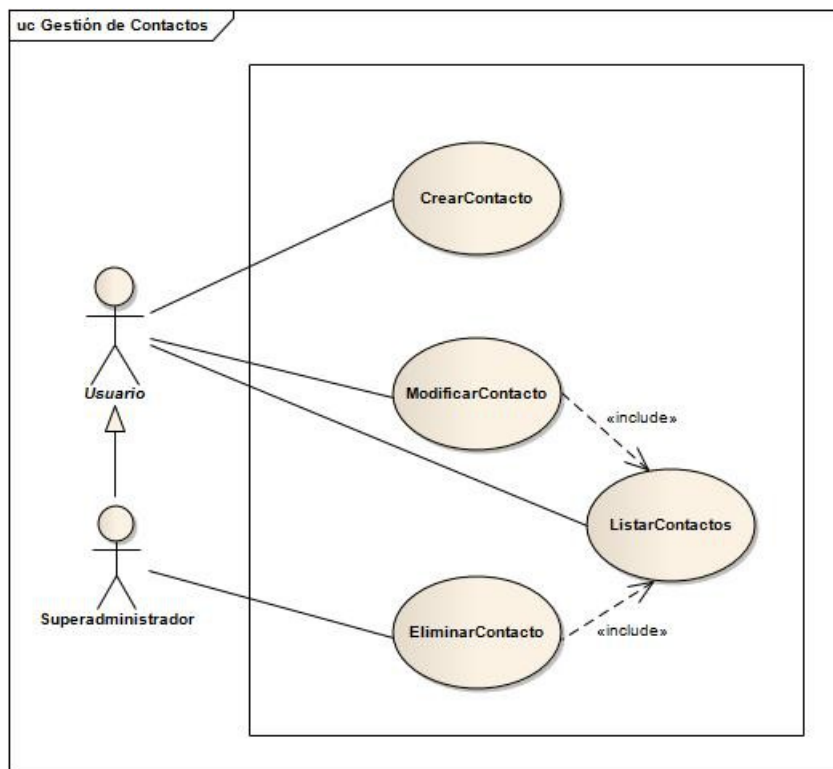
Acciones de los actores	Respuesta del sistema
1. El usuario selecciona la empresa a eliminar del listado de empresas.	
	2. El sistema muestra al usuario el formulario de confirmación de eliminación con los datos de la empresa a eliminar.
3. El usuario pulsa el botón de confirmación de borrado.	
	4. El sistema comprueba que la empresa no esté asignado a ningún evento.
	5. El sistema elimina la empresa.
	6. El sistema muestra al usuario el listado de empresas actualizado y el mensaje de resultado de la operación.

Escenario alternativo:

4a. La empresa está asignada a eventos.

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de error informando de que la empresa está asignada a algún evento.

3.1.5 Gestión de contactos



3.1.5.1 Crear contacto

Actores: Usuario (Superadministrador, Administrador, Editor, Autorizador)

Tipo: Principal

Descripción: Este caso de uso se corresponde con el proceso de crear un nuevo contacto.

Requisitos: El usuario debe estar autenticado en la aplicación.

Escenario de éxito:

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la operación de creación de contacto.	
	2. El sistema muestra el formulario de introducción de datos del contacto.
3. El usuario introduce los datos del nuevo contacto y los acepta.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema registra el nuevo contacto.
	6. El sistema muestra al usuario el mensaje de resultado de la aplicación.

3.1.5.2 Listar contactos

Actores: Usuario (Superadministrador, Administrador, Editor, Autorizador)

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de mostrar el listado de contactos.

Requisitos: El usuario debe estar autenticado en la aplicación.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la opción de mostrar el listado de contactos.	
	2. El sistema consulta los contactos dados de alta en la aplicación.
	3. El sistema muestra al usuario el listado de contactos. La aplicación debe mostrar principalmente los datos de contacto de la persona.

Escenario alternativo:

3a. No se han dado de alta contactos en el sistema.

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de información de no existencia de contactos.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

3b. Filtrado del listado de contactos mostrado.

Acciones de los actores	Respuesta del sistema
1. El usuario introduce los valores de los filtros de búsqueda que desea aplicar sobre el listado de contactos y pulsa el botón buscar.	
	2. El sistema actualiza el listado de contactos aplicando los filtros de búsqueda.

3c. Cancelación de filtros de búsqueda de contactos.

Acciones de los actores	Respuesta del sistema
1. El usuario pulsa el botón de limpiar filtros de búsqueda.	
	2. El sistema restaura el listado de contactos.

3.1.5.3 *Modificar contacto*

Actores: Usuario (Superadministrador, Administrador, Editor, Autorizador)

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de modificar un contacto.

Requisitos: El usuario debe estar autenticado en la aplicación y acceder a la opción de listado de contactos.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el contacto a modificar del listado de contactos.	
	2. El sistema muestra al usuario el formulario de edición de datos de contacto.
3. El usuario modifica los datos del contacto que considere pertinentes.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema actualiza los datos del contacto.
	6. El sistema muestra al usuario el listado de contactos actualizado y el mensaje de resultado de la operación.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

3.1.5.4 Eliminar contacto

Actores: Superadministrador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de eliminar un contacto.

Requisitos: El usuario debe estar autenticado en la aplicación y acceder a la opción de listado de contactos. El usuario debe tener rol de superadministrador de la aplicación.

Escenario de éxito:

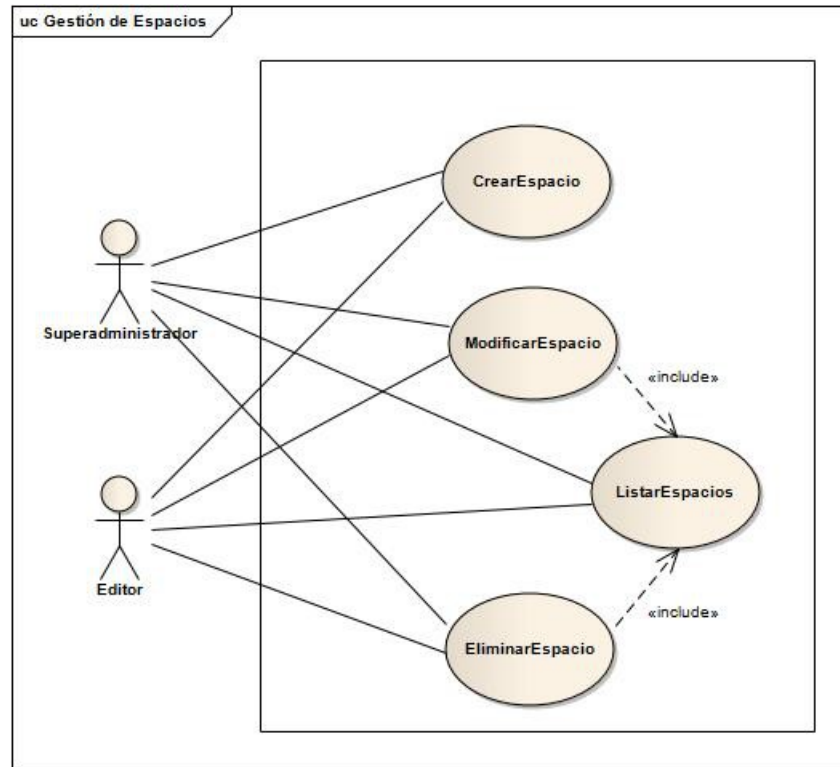
Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el contacto a eliminar del listado de contactos.	
	2. El sistema muestra al usuario el formulario de confirmación de eliminación con los datos del contacto a eliminar.
3. El usuario pulsa el botón de confirmación de borrado.	
	4. El sistema comprueba que el contacto no esté asignado a ningún evento.
	5. El sistema elimina el contacto.
	6. El sistema muestra al usuario el listado de contactos actualizado y el mensaje de resultado de la operación.

Escenario alternativo:

4a. El contacto está asignado a eventos.

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de error informando de que el contacto está asignado a algún evento.

3.1.6 Gestión de espacios



3.1.6.1 Crear espacio

Actores: Superadministrador, Editor

Tipo: Principal

Descripción: Este caso de uso se corresponde con el proceso de crear un espacio.

Requisitos: El usuario debe estar autenticado en la aplicación.

Escenario de éxito:



Gestión de espacios y eventos culturales (GEEC)

Proyecto de fin de carrera

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la operación de creación de espacio.	
	2. El sistema muestra el formulario de introducción de datos del espacio.
3. El usuario introduce los datos del nuevo espacio y los acepta. En el caso del superadministrador se deberá seleccionar también el grupo al que pertenece el espacio.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema registra el nuevo espacio.
	6. El sistema muestra al usuario el mensaje de resultado de la aplicación.

3.1.6.2 Listar espacio

Actores: Superadministrador, Editor

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de mostrar el listado de espacios.

Requisitos: El usuario debe estar autenticado en la aplicación.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la opción de mostrar el listado de espacios.	
	2. El sistema consulta los espacios existentes en la aplicación. En el caso del superadministrador buscará todos y en caso contrario los pertenecientes a su grupo.
	3. El sistema muestra al usuario el listado de espacios.

Escenario alternativo:

3a. No se han dado de alta espacios en el sistema o en el grupo.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de información de no existencia de espacios.

3b. Filtrado del listado de espacios mostrado.

Acciones de los actores	Respuesta del sistema
1. El usuario introduce los valores de los filtros de búsqueda que desea aplicar sobre el listado de espacios y pulsa el botón buscar.	
	2. El sistema actualiza el listado de espacios aplicando los filtros de búsqueda.

3c. Cancelación de filtros de búsqueda de empleados.

Acciones de los actores	Respuesta del sistema
1. El usuario pulsa el botón de limpiar filtros de búsqueda.	
	2. El sistema restaura el listado de espacios.

3.1.6.3 *Modificar espacio*

Actores: Superadministrador, Editor

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de modificar los datos de un espacio.

Requisitos: El usuario debe estar autenticado en la aplicación y acceder a la opción de listado de espacios.

Escenario de éxito:



Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el espacio a modificar del listado de espacios.	
	2. El sistema muestra al usuario el formulario tabulado de edición de datos de espacio.
3. El usuario modifica o añade los datos del espacio que considere pertinentes.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema actualiza los datos del espacio.
	6. El sistema muestra al usuario el listado de espacios actualizado y el mensaje de resultado de la operación.

3.1.6.4 Eliminar espacio

Actores: Superadministrador, Editor

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de eliminar un espacio. Debido a la necesidad de mantener un histórico de eventos, los espacios no podrán ser eliminados debido a la relación de los eventos con los espacios. Cuando se elimine un espacio se pasará el mismo al estado deshabilitado.

Requisitos: El usuario debe estar autenticado en la aplicación y acceder a la opción de listado de espacios.

Escenario de éxito:



Gestión de espacios y eventos culturales (GEEC)

Proyecto de fin de carrera

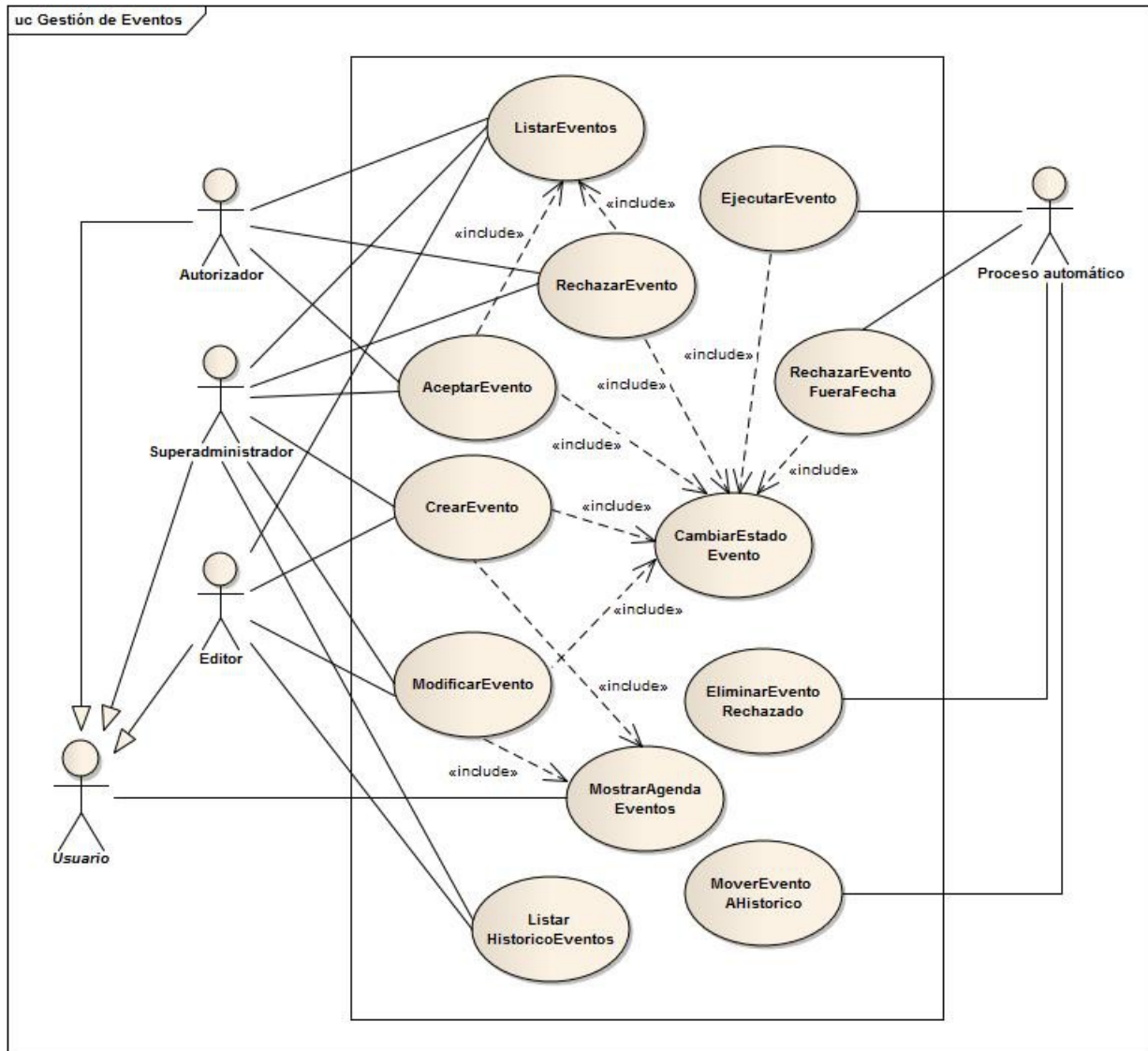
Acciones de los actores	Respuesta del sistema
1. El usuario selecciona el espacio a eliminar del listado de espacios.	
	2. El sistema muestra al usuario el formulario de confirmación de eliminación con los datos del espacio a eliminar.
3. El usuario pulsa el botón de confirmación de borrado.	
	4. El sistema comprueba que el espacio no esté asignado a ningún evento pendiente de ejecutar.
	5. El sistema elimina el espacio actualizando su estado a deshabilitado.
	6. El sistema muestra al usuario el listado de espacios actualizado y el mensaje de resultado de la operación.

Escenario alternativo:

4a. El espacio está asignado a eventos pendientes de ejecutar.

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de error informando de que el espacio está asignado a algún evento.

3.1.7 Gestión de eventos



3.1.7.1 Agenda de eventos

Actores: Usuario (Superadministrador, Administrador, Editor, Autorizador)

Tipo: Principal

Descripción: Este caso de uso se corresponde con el proceso de consultar la agenda de eventos programados.

Requisitos: El usuario debe estar autenticado en la aplicación.

Escenario de éxito:



Acciones de los actores	Respuesta del sistema
1. El usuario accede a la aplicación o selecciona la opción de mostrar agenda de eventos.	
	2. El sistema muestra la agenda de eventos paginándolos de semana en semana.

3.1.7.2 Crear evento

Actores: Superadministrador, Editor

Tipo: Principal

Descripción: Este caso de uso se corresponde con el proceso de crear un evento asociado a un espacio y una fecha. El evento se creará en estado 'Pendiente de autorización'.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el perfil de superadministrador o editor y acceder a la agenda de eventos, ya sea al iniciar sesión en la aplicación o al seleccionar la opción de mostrar la agenda de eventos.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario selecciona la fecha de la agenda en la cual quiere crear el evento.	
	2. El sistema muestra el formulario de creación de evento.
3. El usuario introduce los datos del nuevo evento y los acepta.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema registra el nuevo evento y lo establece en el estado 'Pendiente'.
	6. El sistema muestra al usuario el mensaje de resultado de la aplicación y actualiza la agenda de eventos.

3.1.7.3 Modificar evento

Actores: Superadministrador, Editor

Tipo: Principal

Descripción: Este caso de uso se corresponde con el proceso de modificación de un evento asociado a un espacio y una fecha. Se podrá modificar un evento siempre que no haya sido ejecutado (es decir

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

anterior a la fecha de ejecución) y se pasará al estado 'Pendiente de autorización'.

Requisitos: El usuario debe estar autenticado en la aplicación, tener el perfil de superadministrador o editor y acceder a la agenda de eventos, ya sea al iniciar sesión en la aplicación o al seleccionar la opción de mostrar la agenda de eventos.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario selecciona en la agenda de eventos, el evento a modificar.	
	2. El sistema muestra el formulario de modificación del evento.
3. El usuario introduce los nuevos datos del evento y los acepta.	
	4. El sistema comprueba que se han introducido los datos obligatorios.
	5. El sistema actualiza el evento y lo establece en el estado 'Pendiente'.
	6. El sistema muestra al usuario el mensaje de resultado de la aplicación y actualiza la agenda de eventos.

3.1.7.4 Listar eventos

Actores: Superadministrador, Editor, Autorizador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de mostrar el listado de eventos de la aplicación donde se podrán realizar búsquedas de los mismos aplicando filtros.

Requisitos: El usuario debe estar autenticado en la aplicación.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la opción de mostrar el listado de eventos.	
	2. El sistema consulta los eventos existentes en la aplicación para el grupo al que pertenezca el usuario o todos los eventos en el caso del superadministrador.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

Escenario alternativo:

2a. No se han dado de alta eventos en el grupo o en el sistema.

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de información de no existencia de eventos.

2b. Filtrado del listado de eventos mostrado.

Acciones de los actores	Respuesta del sistema
1. El usuario introduce los valores de los filtros de búsqueda que desea aplicar sobre el listado de eventos y pulsa el botón buscar.	
	2. El sistema actualiza el listado de eventos aplicando los filtros de búsqueda.

2c. Cancelación de filtros de búsqueda de eventos.

Acciones de los actores	Respuesta del sistema
1. El usuario pulsa el botón de limpiar filtros de búsqueda.	
	2. El sistema restaura el listado de eventos.

3.1.7.5 Listar del histórico de eventos

Actores: Superadministrador, Editor, Autorizador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de mostrar el listado del histórico de eventos ejecutados de la aplicación donde se podrán realizar búsquedas de los mismos aplicando filtros.

Requisitos: El usuario debe estar autenticado en la aplicación.

Escenario de éxito:

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la opción de mostrar el listado del histórico de eventos ejecutados.	
	2. El sistema consulta los eventos ejecutados existentes en la aplicación para el grupo al que pertenezca el usuario o todos los eventos en el caso del superadministrador.

Escenario alternativo:

2a. No se han ejecutado eventos en el grupo o en el sistema.

Acciones de los actores	Respuesta del sistema
	1. El sistema muestra al usuario el mensaje de información de no existencia de eventos ejecutados.

2b. Filtrado del listado de eventos ejecutados mostrado.

Acciones de los actores	Respuesta del sistema
1. El usuario introduce los valores de los filtros de búsqueda que desea aplicar sobre el listado de eventos ejecutados y pulsa el botón buscar.	
	2. El sistema actualiza el listado de eventos ejecutados aplicando los filtros de búsqueda.

2c. Cancelación de filtros de búsqueda de eventos ejecutados.

Acciones de los actores	Respuesta del sistema
1. El usuario pulsa el botón de limpiar filtros de búsqueda.	
	2. El sistema restaura el listado de eventos ejecutados.

3.1.7.6 *Aceptar evento*

Actores: Superadministrador, Autorizador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de aceptar un evento que se encuentre en estado 'Pendiente de autorización'.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

Requisitos: El usuario debe estar autenticado en la aplicación con el rol de superadministrador o autorizador y acceder al listado de eventos.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El usuario accede a la opción de mostrar el listado de eventos.	
	2. El sistema consulta los eventos del grupo del usuario o todos los eventos en caso del superadministrador.
	3. El sistema muestra al usuario el listado de eventos pendientes.
4. El usuario selecciona el botón de 'Aceptar evento'.	
	5. El sistema muestra al usuario el listado de eventos actualizado y el mensaje de resultado de la operación.

3.1.7.7 Rechazar evento

Actores: Superadministrador, Autorizador

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de aceptar un evento que se encuentre en estado 'Pendiente de autorización' o 'Aceptado'.

Requisitos: El usuario debe estar autenticado en la aplicación con el rol de superadministrador o autorizador y acceder al listado de eventos.

Escenario de éxito:



Acciones de los actores	Respuesta del sistema
1. El usuario accede a la opción de mostrar el listado de eventos.	
	2. El sistema consulta los eventos del grupo del usuario o todos los eventos en caso del superadministrador.
	3. El sistema muestra al usuario el listado de eventos pendientes.
4. El usuario selecciona el botón de 'Rechazar evento'.	
	5. El sistema muestra al usuario el listado de eventos actualizado y el mensaje de resultado de la operación.

3.1.7.8 Ejecutar evento

Actores: Proceso automático

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de cambio del estado de un evento 'Aceptado' al estado 'Ejecutado' una vez superada la fecha en que se haya programado su ejecución.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El proceso comprueba que se trate de la fecha programada para el lanzamiento de la tarea y se ejecuta la misma.	
	2. Se consultan los eventos cuya fecha de ejecución sea superior a la fecha actual y su estado sea 'Aceptado'.
	3. Se cambia el estado de los eventos cuya fecha de ejecución sea superior a la fecha actual al estado 'Ejecutado'.

3.1.7.9 Rechazar evento fuera de fecha

Actores: Proceso automático

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de cambio del estado de un evento 'Pendiente' al estado 'Rechazado' una vez que la fecha de programación del evento se encuentra a 15

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

días o menos de la fecha actual y el evento no se ha pasado al estado aceptado.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El proceso comprueba que se trate de la fecha programada para el lanzamiento de la tarea y se ejecuta la misma.	
	2. Se consultan los eventos cuya fecha de ejecución sea inferior a 15 días de la fecha actual y su estado sea 'Pendiente'.
	3. Se cambia el estado dichos eventos al estado 'Rechazado'.

3.1.7.10 *Eliminar evento rechazado*

Actores: Proceso automático

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de borrado de los eventos que se encuentren en estado 'Rechazado' y cuya fecha de ejecución sea posterior a la fecha actual.

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El proceso comprueba que se trate de la fecha programada para el lanzamiento de la tarea y se ejecuta la misma.	
	2. Se consultan los eventos cuya fecha de ejecución sea superior a la fecha actual y su estado sea 'Rechazado'.
	3. Se eliminan del sistema dichos eventos.

3.1.7.11 *Mover eventos al histórico*

Actores: Proceso automático

Tipo: Principal

Descripción: Este caso de uso se corresponde con la operación de mover los eventos que se encuentren en estado 'Ejecutado' al histórico de eventos, una vez que su fecha de ejecución es posterior a un año desde la fecha actual.

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

Escenario de éxito:

Acciones de los actores	Respuesta del sistema
1. El proceso comprueba que se trate de la fecha programada para el lanzamiento de la tarea y se ejecuta la misma.	
	2. Se consultan los eventos cuya fecha de ejecución sea superior a un año a partir de la fecha actual y su estado sea 'Ejecutado'.
	3. Se mueven dichos eventos al histórico de eventos.

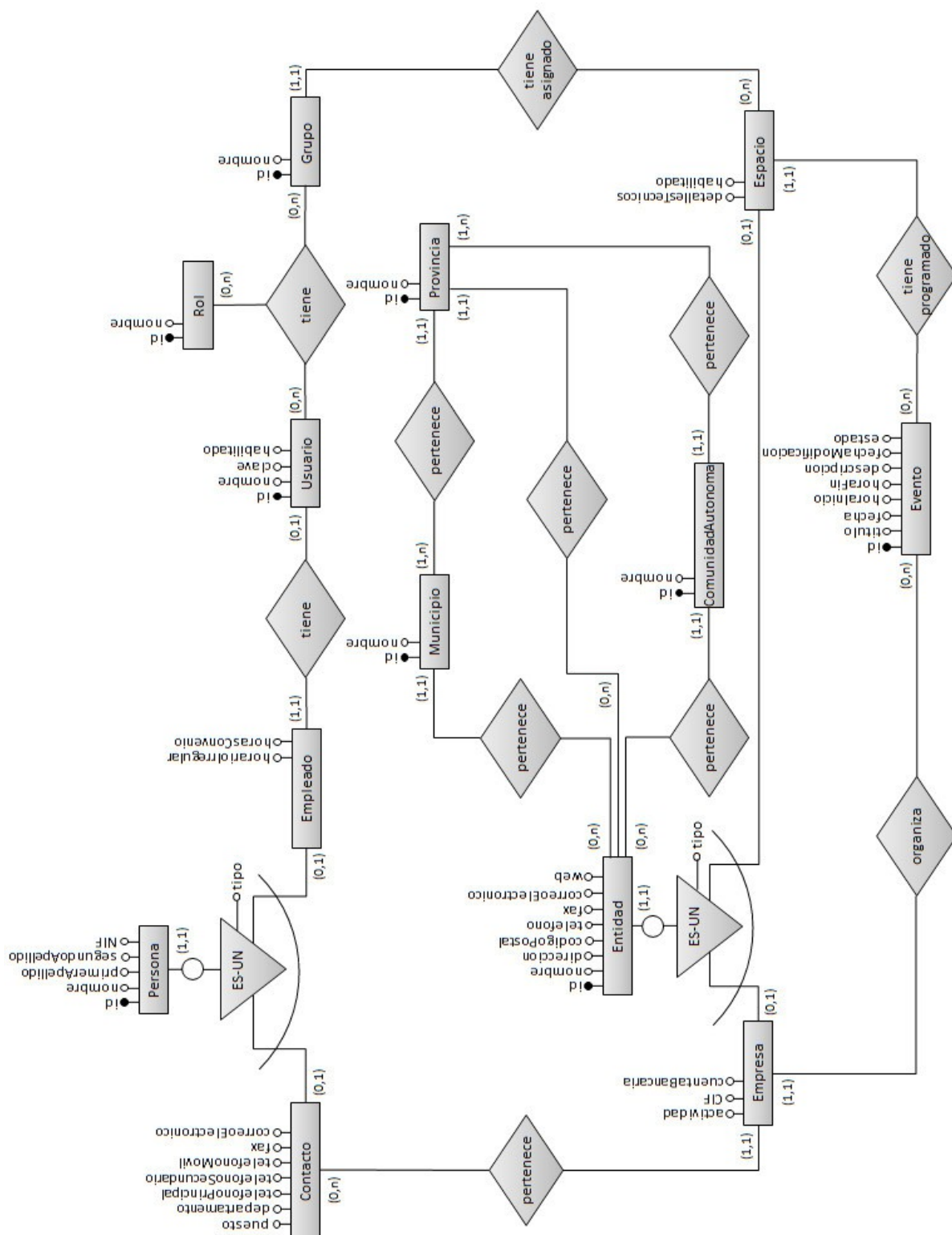
3.2 Modelo de datos

En este apartado identificaremos las entidades y sus atributos y las relaciones entre las mismas que permitan cubrir los requisitos de datos recogidos durante la toma de requisitos. Para ello se ha desarrollado el Modelo E/R Extendido del sistema que vamos a implementar, a partir del cual se creará posteriormente el diseño físico de base de datos.

3.2.1 Modelo conceptual de datos

Para obtener el modelo conceptual se utilizará un Modelo E/R Extendido para poder representar los datos que se procesarán en el sistema, independientemente de cualquier tipo de restricción tecnológica que se pueda aplicar.

La siguiente imagen muestra el diagrama E/R Extendido del modelo conceptual de datos que se va a utilizar para el desarrollo del sistema de gestión de eventos y espacios culturales. Este diagrama muestra las entidades acerca de las cuales el sistema necesita almacenar información, con los atributos relativos a cada una de ellas, y las relaciones o correspondencias entre las mismas.





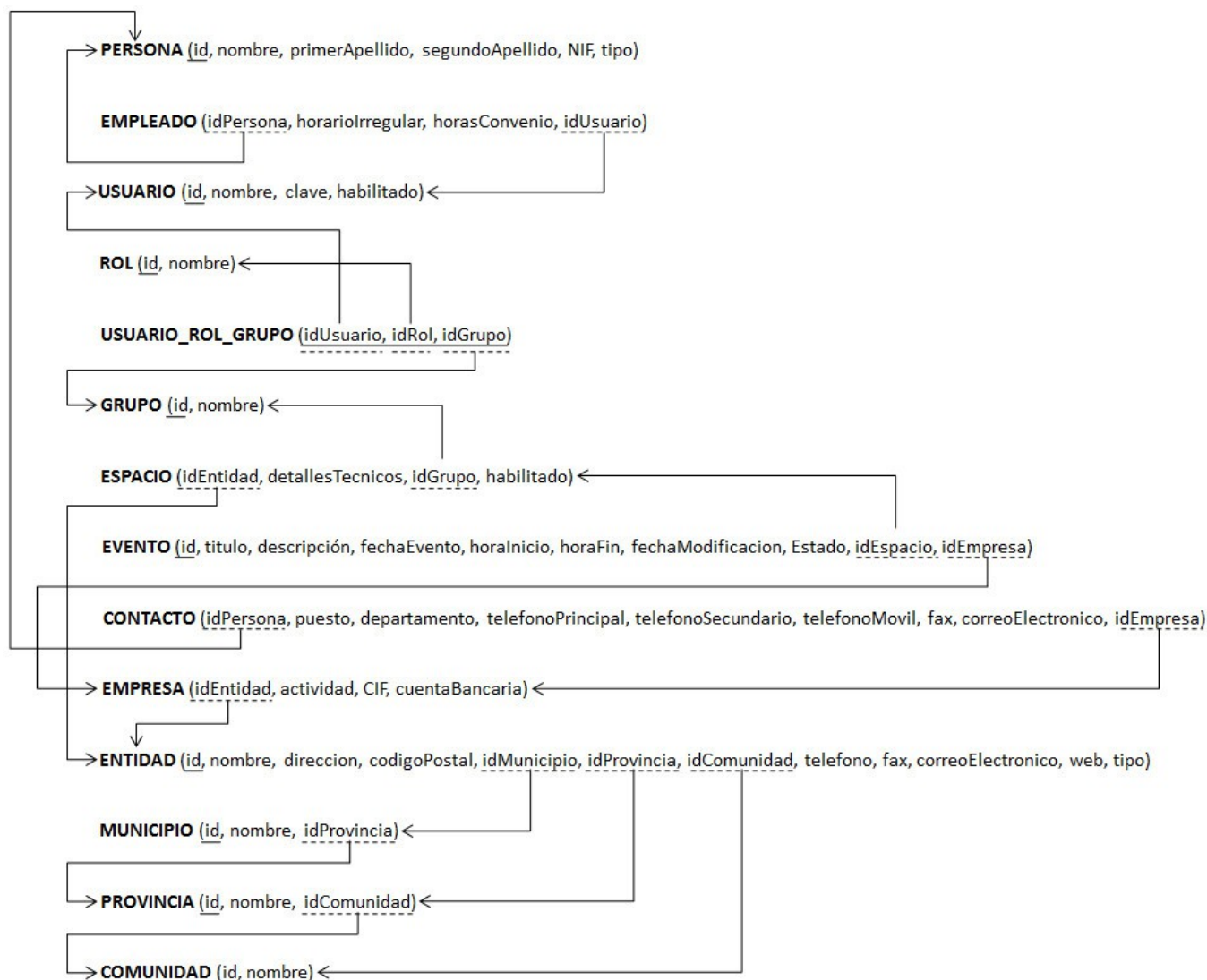
Las características más importantes del modelo conceptual anterior son las siguientes:

- Se ha creado el supertipo de entidad **Persona**, en el cual se almacenará la información común relativa a las entidades específicas de **Contacto** y **Empleado**. El discriminante será el atributo **tipo** que podrá tomar los valores 'Contacto' o 'Empleado'.
- Se ha creado el supertipo de entidad **Entidad**, en el cual se almacenará la información común relativa a las entidades específicas de **Empresa** y **Espacio**. El discriminante será el atributo **tipo** que podrá tomar los valores 'Empresa' o 'Espacio'.
- Aparece una relación ternaria entre las entidades **Usuario**, **Rol** y **Grupo**, que no se puede descomponer, pues no recogería la semántica necesaria para la aplicación, donde un usuario puede tener uno o varios roles en uno o varios grupos.
- Aunque las entidades **Municipio**, **Provincia** y **Comunidad** están relacionados entre sí, y se podría considerar relacionar con **Entidad** únicamente el municipio, esto no recogería la semántica del sistema, donde una entidad puede almacenar o no el municipio, la provincia y la comunidad como parte de su dirección.
- No se han incluido las entidades encargadas de almacenar la información relativa al histórico de eventos (ya que tendrá los mismos campos que la tabla de eventos y la misma relación con la tabla espacio), ni las trazas de operación, ya que no están definidas por el usuario como requisitos de almacenamiento, dejando a la elección del desarrollo el modo en que se quiere almacenar dicha información.

Una vez comprobado que el diseño está normalizado y que cubre todas las necesidades impuestas por los requisitos definidos por el usuario, se generará el modelo lógico de datos a partir del mismo y se generará la representación física de los mismos en el sistema gestor de base de datos.

3.2.2 Modelo lógico estándar

A partir del modelo conceptual definido en el apartado anterior se genera el modelo lógico estándar, transformando las entidades en tablas y resolviendo las interrelaciones N:M y 1:N. El modelo lógico resultante es el siguiente:



Las reglas aplicadas para resolver el modelo lógico estándar anterior son:

- Se ha solucionado la generalización de **Persona**, **Empleado** y **Contacto** creando una tabla independiente para cada subtipo, ya que proporciona mayor semántica y hay bastantes atributos distintos en los subtipos. Las tablas específicas poseerán como atributo el **idPersona** que relacionará cada uno de los registros con uno de la tabla de personas.
- Se ha solucionado la generalización de **Entidad**, **Empresa** y **Espacio** creando una tabla independiente para cada subtipo porque proporciona mayor semántica y hay bastantes atributos distintos en los subtipos. Las tablas específicas poseerán como atributo el **idEntidad** que relacionará cada uno de los registros con uno de la tabla de personas.
- Se ha solucionado el problema de la relación compleja ternaria **Usuario**, **Rol**, **Grupo** aplicando la regla general para crear la tabla **Usuario_Rol_Grupo** compuesta con las claves primarias de



las tres tablas relacionadas y que tendrá los atributos de borrado y actualización en cascada.

- Se ha resuelto la relación entre **Grupo** y **Espacio** de tipo 0:N, mediante la técnica de propagación de clave de la tabla que tiene cardinalidad máxima 1 a la que tiene N, es decir, se incluye el atributo **idGrupo** en la tabla de espacios. Además se le asignarán los atributos de borrado restringido y actualización en cascada, ya que no se debe eliminar un grupo mientras tenga espacios asignados.
- Se han resuelto las relaciones entre **Evento**, **Espacio** y **Empresa** de tipo 0:N, mediante la técnica de propagación de clave de la tabla que tiene cardinalidad máxima 1 a la que tiene N. Se incluyen los atributos **idEspacio** e **idEmpresa** en la tabla de eventos. Además con relación a los espacios, no se podrá eliminar un espacio siempre que haya participado en algún evento y se utilizará la misma tabla como histórico de espacios. Con respecto a la empresas, no se permitirá su eliminación mientras que haya participado en algún evento y se utilizará la misma tabla como histórico de empresas. Los eventos sí que podrán eliminarse pasado un tiempo establecido por el cliente después de su ejecución, pero se mantendrán en la tabla de histórico de eventos.
- Se han resuelto las relaciones entre las tablas **Municipio**, **Poblacion** y **Comunidad** con **Entidad**, de tipo 0:N, mediante la técnica de propagación de clave de la tabla que tiene cardinalidad máxima 1 a la que tiene N, es decir, se incluyen los atributos **idMunicipio**, **idPoblacion** e **idComunidad** en la tabla de entidades. Además se le asignarán los atributos de borrado restringido y actualización en cascada a estas tablas, ya que no se debe eliminar un municipio, población o comunidad mientras tenga entidades asignadas.

Una vez establecidos estas soluciones de diseño, se crea el diseño lógico específico de la base de datos, en relación con el sistema gestor de base de datos elegido para la implementación de base de datos, MySQL.

3.2.3 Modelo físico

A partir del modelo lógico estándar establecido anteriormente se genera el modelo físico para el sistema gestor de bases de datos seleccionado para la implementación del proyecto, MySQL.

Para el desarrollo del mismo se utilizarán las siguientes convenciones de nombrado:

- Los nombres de las tablas irán en singular y en caso de nombres compuestos, se separarán utilizando el carácter subrayado (_).
- Los identificadores de las tablas de denominarán con la palabra **id**, y cuando se haga referencia a los mismos como claves ajenas se utilizará **id** más el nombre de la tabla a la que haga referencia (ej: **idEntidad**).



- Los índices que hagan referencia a claves primarias de la tabla utilizarán la siguiente nomenclatura:

pk_<tabla a la que referencia>_<tabla del índice>.

- Los índices que hagan referencia a restricciones de claves ajenas de la tabla utilizarán la siguiente nomenclatura:

fk_<tabla a la que referencia>_<tabla del índice>.

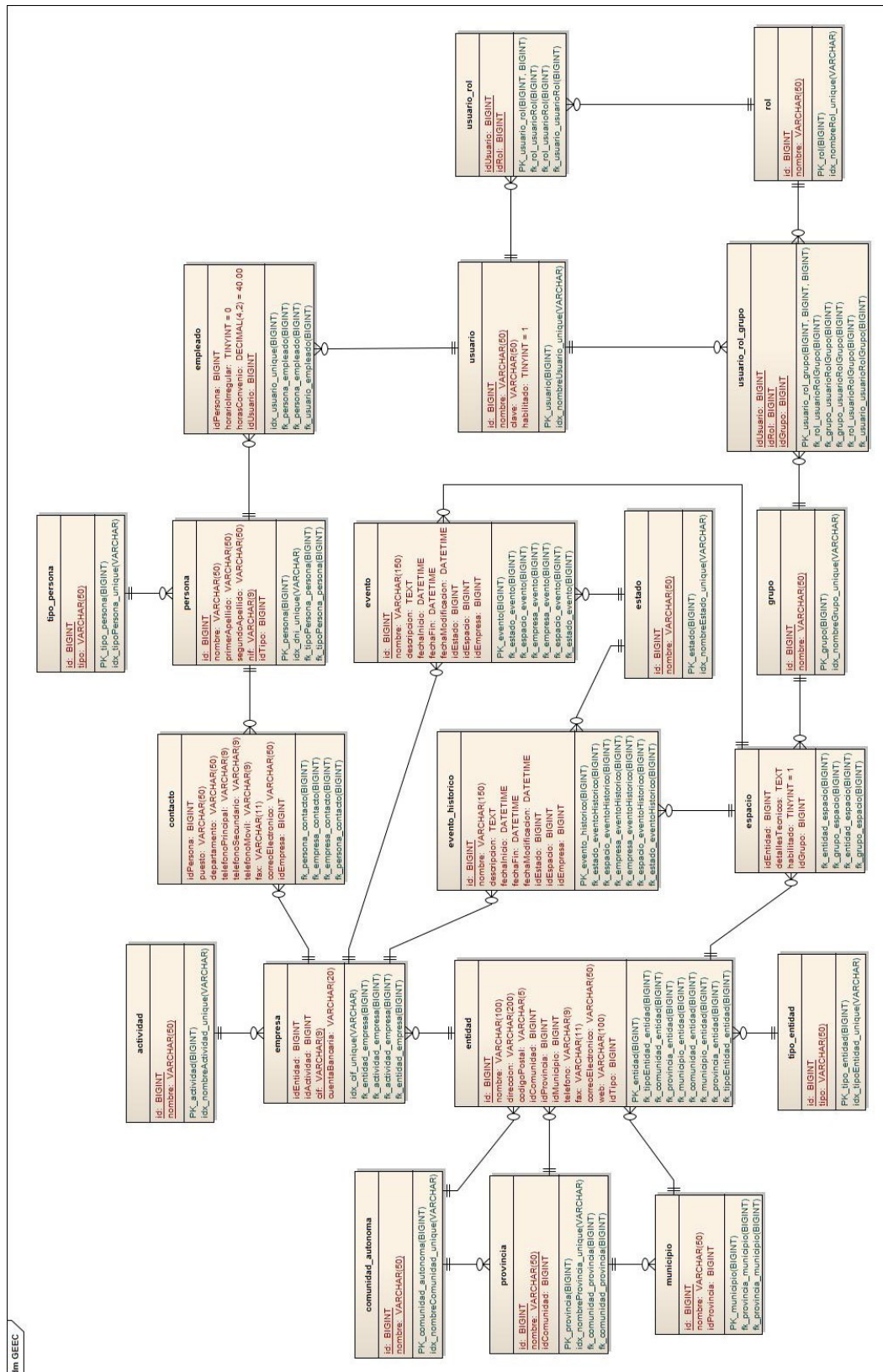
- Los índices que hagan referencia a restricciones de clave única utilizarán la siguiente nomenclatura:

idx_<nombre del campo (y opcionalmente el nombre de la tabla)>_<tipo de restricción>

Además, debido a los frameworks utilizados para el desarrollo del sistema, se han de tener en cuenta las siguientes restricciones a la hora de generar el modelo físico de base de datos:

- El tipo utilizado para las columnas índice será **BIGINT**, aunque esto no es obligatorio, debido a que el estándar de persistencia de datos utilizado, JPA-HIBERNATE, gestiona mejor este tipo de datos para secuencias auto-incrementales.
- Se han creado tablas propias para los tipos admitidos por los atributos discriminantes de las tablas **persona** y **entidad**, ya que el framework de persistencia de datos facilita su utilización realizando el diseño de esta manera. Además, debido a que se trata de un proyecto evolutivo, diseñar el modelo de datos de esta forma facilitará la extensión del mismo cuando sea necesario.
- Se han creado tablas propias para los atributos **actividad** y **estado**. Esto se debe también a que, al tratarse de un proyecto evolutivo, diseñar el modelo de datos de esta forma permitirá añadir nuevos tipos de actividad y de estado de forma dinámica sin tener que modificar el código de la aplicación (o modificándolo de forma leve).
- Aparece la tabla **usuario_rol**, que no aparecía en el modelo lógico. Aunque se podría evitar la inclusión de esta tabla, realmente facilita la tarea de integración del framework de seguridad elegido para la creación de la aplicación, **Spring Security**, ya que dicho framework trabaja sobre relaciones de usuario y rol, y necesita de un esquema de base de datos propio que se ha modificado levemente para incluir las restricciones del proyecto, pero manteniendo su estructura inicial. Por ello aparece esta tabla. Además, se evita el tener que definir el registro del superadministrador para cada registro de nuevo grupo que se incluya en la aplicación, ya que se podrá utilizar el rol de superadministrador sin tener en cuenta que esté asignado a ningún grupo, utilizando esta tabla. También permitirá tener más usuarios con roles genéricos independientemente del grupo, aunque de momento no es un requisito de sistema.

Por tanto el modelo físico de base de datos quedará como se muestra en la siguiente imagen:

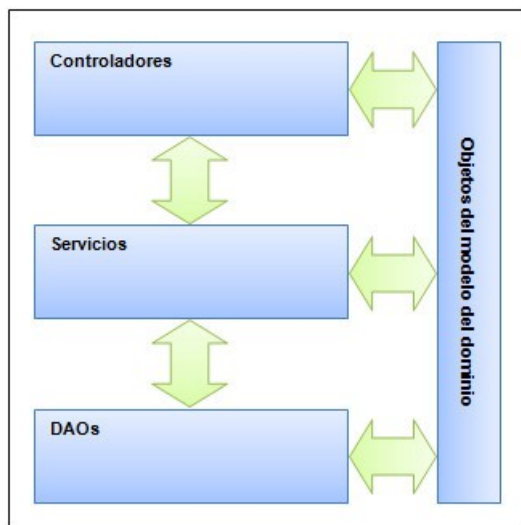


El script de base de datos para la creación del modelo físico anterior se incluirá en el apartado **7.1 del Anexo A** junto con los scripts necesarios para los datos de inicialización y de prueba.

3.3 Diagramas de clases (estructurales)

El diagrama de clases define las clases y asociaciones de las mismas que implementan el sistema software que se va a construir, representando la estructura y el comportamiento de cada uno de los objetos.

Para la creación de la arquitectura de desarrollo solicitada como requisito, se han utilizado patrones de diseño que permiten mejorar la reutilización y estructuración del código. En concreto, los más interesantes son el patrón **MVC** (model view controller o modelo vista controlador), el patrón y **DAO** (data access object u objetos de acceso a datos), los cuales definirán que la arquitectura a desarrollar se divida en distintas capas cada una de las cuales se encargará de realizar una tarea dentro del conjunto global de la aplicación. Por tanto, se dividirán los diagramas de clases en varios grupos, atendiendo a cada una de las capas mostradas en el siguiente diagrama:



Las clases del **modelo del dominio** son las clases que modelan el sistema que a implementar y se corresponden con las distintas tablas de base de datos. Estas clases interactúan con las clases de todas las capas, pues son las clases que recogen la información de base de datos y la presentan las mismas en las páginas web que se mostrarán al usuario.

Las clases de la capa de **Controladores** son las que recibirán las peticiones del usuario desde las páginas web y redirigirán las peticiones a las clases de servicio que contienen la lógica de negocio de la aplicación. Realizan también todas las operaciones o transformaciones relativas a la presentación.

Las clases de la capa de **Servicios** son las que realizan la lógica de negocio propiamente dicha. Es



decir, realizan todos los cálculos u operaciones específicos del ámbito de negocio que cubre la aplicación, en este caso, la gestión de espacios y eventos culturales. También se encargarán de redirigir las consultas de base de datos a los respectivos DAOs que sean necesarios para realizar una operación de negocio completa.

Por último, las clases de la capa de **DAOs** son las que actúan como intermediarias de los servicios de lógica de negocio y la base de datos, y contienen todos los métodos necesarios para cargar los objetos del modelo con los datos de la base de datos.

Aunque la aplicación tiene muchas más clases de apoyo, el cuerpo principal de la misma se encuentra en estas capas, por tanto sólo se incluirán los diagramas relativos a las mismas.

Además, se han seguido las reglas del desarrollo orientado a interfaces (referido a clases abstractas o interfaces) para las capas de servicios y DAOs, lo cual permite que al utilizar interfaces en las referencias a los objetos para cada una de las clases del sistema, se consiga desacoplar las distintas capas de las que se compone facilitando la reutilización de código y el mantenimiento y evolución del mismo. Por tanto, por cada clase de las capas de servicios y DAOs tendremos su interface de definición y su clase o clases de implementación, ya que para una única interface se podrán tener tantas implementaciones como sean necesarias.

En este apartado se incluirán los diagramas de clases de la fase de análisis estructural del proyecto, es decir, se incluirán los diagramas con sus relaciones en relación a la estructura organizativa de la arquitectura definida según la imagen anterior, independientemente de las asociaciones que existan entre las clases de las distintas capas de la misma o de la colaboración que exista entre ellas, ya que esto se analizará en relación de las distintas funcionalidades en el siguiente apartado. Por este motivo, se van a analizar 4 diagramas distintos:

- Diagrama de clases del modelo de dominio: Describirá las clases que deben componer el modelo de dominio que define los objetos de la aplicación, la relación de los mismos y las posibles modificaciones sobre el mismo que se hayan de realizar para incluir los distintos frameworks de desarrollo a utilizar.
- Diagrama de clases de controladores: Describirá las clases que deben componer la capa de controladores del patrón Modelo-Vista-Controlador y que contendrán la lógica relativa a la presentación de datos al usuario.
- Diagrama de clases de servicios: Describirá las clases que deben componer la capa de servicios de lógica de negocio, y que será el código de negocio que permanecerá estático aunque se cambie la presentación de datos.
- Diagrama de clases de acceso a datos: Describirá las clases que deben mapear el acceso a datos con las tablas del modelo de datos diseñado.



La imagen muestra el diagrama de las clases del **modelo de dominio** que representan el sistema que se está implementando.

Estas clases se corresponden con las distintas tablas del modelo de base de datos diseñado para el sistema, y se utilizarán para cargar los datos obtenidos de la misma y transportarlos por las distintas capas de la aplicación, realizar los cálculos necesarios sobre los datos contenidos en las mismas y mostrar los datos al usuario en la capa de presentación.

El único aspecto importante a destacar es la clase `UsuarioRolGrupold`, que aparece como requisito de la tecnología utilizada para implementar la capa de acceso a datos, ya que al tener la tabla **usuario_rol_grupo** una clave primaria compuesta por la clave primaria de otras tres tablas, es necesario crearnos una clase que implemente este tipo de datos.

3.3.2 Diagrama de clases de controladores

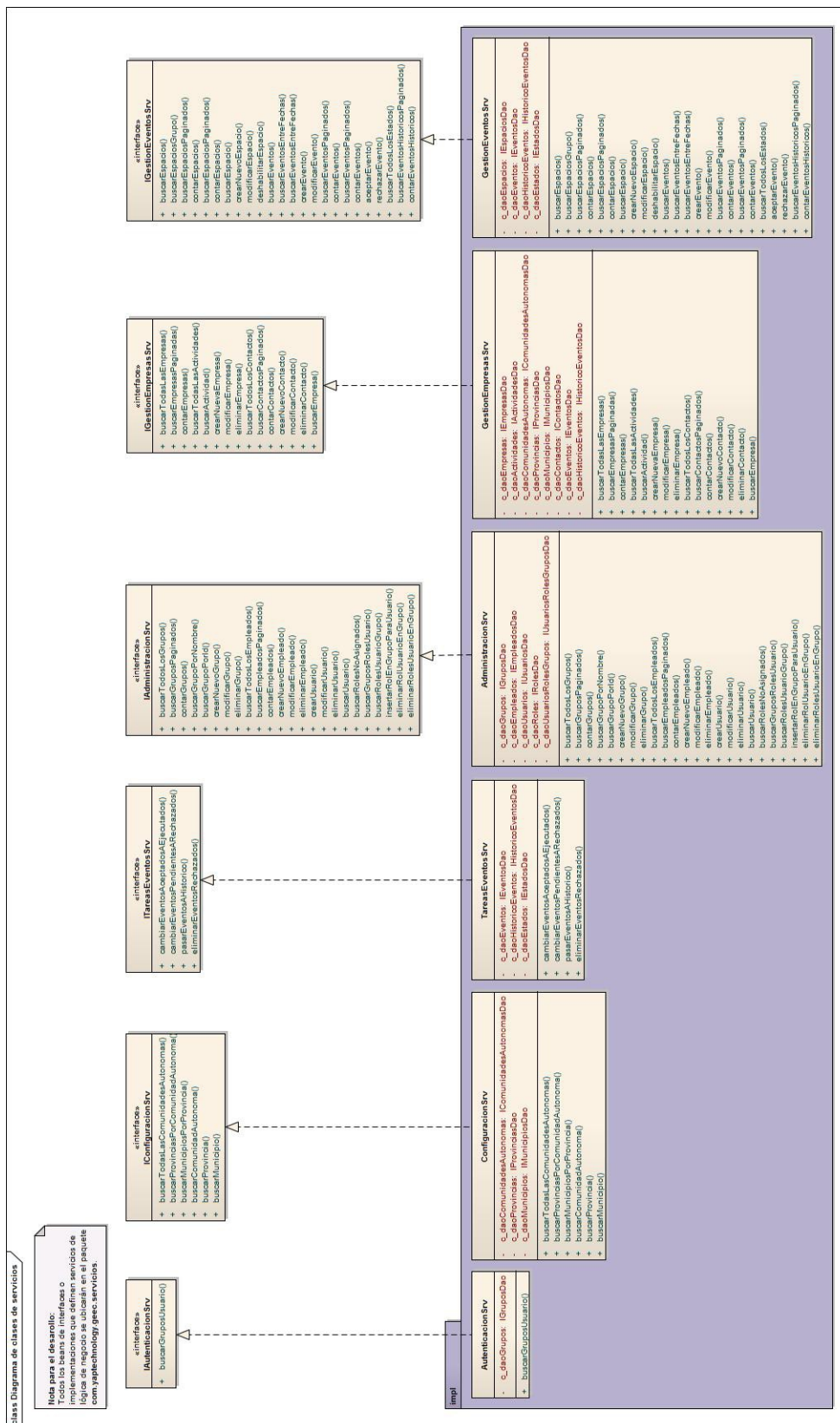
La siguiente imagen muestra el diagrama de clases que implementarán los controladores de la arquitectura Modelo-Vista-Controlador a definir.

Estas clases se contendrán el código de las operaciones que se puedan realizar sobre los datos que se mostrarán en pantalla al usuario o para formatear los mismos y no contendrán código de lógica de negocio, sino únicamente código de respaldo para las páginas web, tecnología que se utilizará para la presentación de datos.

Las clases se agruparán atendiendo a la funcionalidad o grupo de funcionalidades que desempeñen, autenticación, administración, empresas y eventos. Las clases de autenticación contendrán las operaciones que se lanzarán desde las páginas web desde las que el usuario se autenticará en el sistema. Las clases de administración contendrán las operaciones que se lanzarán desde las páginas web que gestionan la administración de la aplicación, es decir, todas las operaciones relacionadas, con la gestión de grupos, empleados y usuarios. Las clases de empresas contendrán las operaciones que se lanzarán desde las páginas de gestión de empresas organizadoras de eventos y contactos de las mismas, y que se compartirán entre todos los grupos de usuarios. Por último, las clases de eventos contendrán las operaciones que se lanzarán desde las páginas de gestión de espacios, eventos y eventos del histórico.



3.3.3 Diagrama de clases de servicios





La imagen anterior muestra el diagrama de clases que definirán e implementarán las clases que contendrán la lógica de negocio de las operaciones de la aplicación.

Las clases que se crearán en esta capa de la aplicación seguirán el desarrollo orientado a interfaces, por lo que se tendrán que crear interfaces que definirán las operaciones que contendrán las implementaciones de las mismas. Esto permitirá que, si cambia el framework o la implementación de la misma, no se tenga que cambiar o modificar la capa de controladores ya que todas las referencias a los objetos de esta capa que se haga por parte de los controladores será sobre las interfaces.

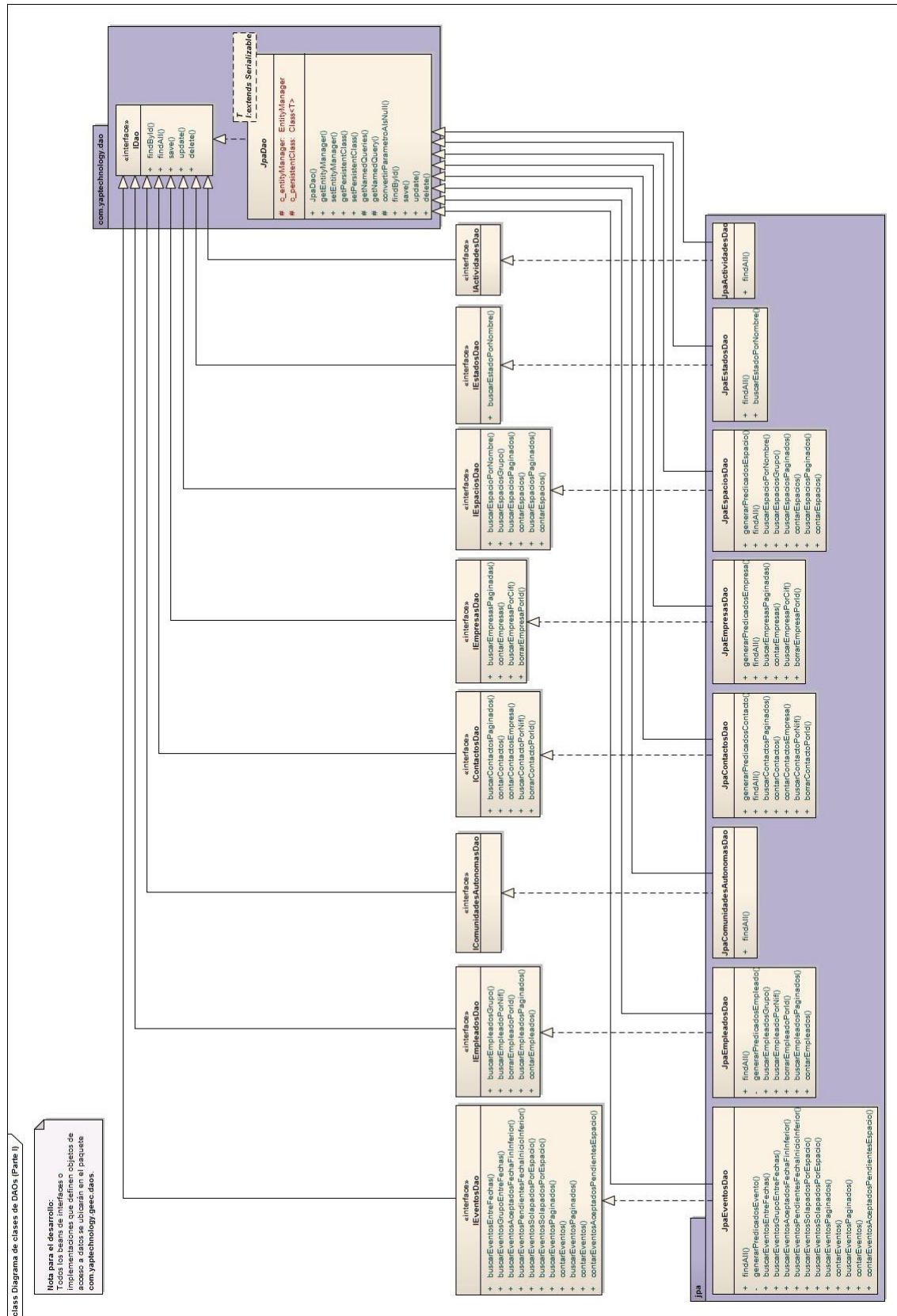
Del mismo modo que en la capa de controladores, se agrupará la lógica de negocios en servicios atendiendo a las funcionalidades que representen, con lo que tendremos un servicio para autenticación, otro servicio para configuración (entendiendo como configuración la gestión de datos estáticos, es decir que no cambiarán nunca en base de datos), un servicio de administración con la lógica de negocio relativa a grupos y usuarios, un servicio de empresas con la lógica de negocio relativa a empresas y contactos, un servicio de eventos con la lógica de negocio relativa a espacios y eventos, y por último un servicio relativo a las tareas automatizadas relativas a los eventos.

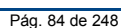
3.3.4 Diagrama de clases de acceso a datos

La siguiente imagen muestra el diagrama de las clases que definirán e implementarán las clases de acceso a datos o DAOs.

Las clases que se crearán en esta capa de la aplicación también seguirán el desarrollo orientado a interfaces. Se crearán interfaces que definirán las operaciones que contendrán las operaciones de las mismas, lo cual permitirá que si se cambia el framework o implementación de acceso a datos no se tenga que modificar la capa de servicios, ya que todas las referencias a los objetos de la capa de acceso a datos se realizarán mediante las interfaces.

En esta capa de la arquitectura se definirán una interface y una implementación básicas que habrán de extender e implementar cada una de las clases de acceso a datos, y que definirán las operaciones básicas sobre los mismos, que son IDao y JpaDaolmpl, ya que se utilizará JPA como framework de acceso a datos en la aplicación. También se crearán una interface que definirá los métodos de acceso a datos específicos de cada uno de los objetos y una implementación para cada una de ellas para el framework JPA.







3.4 Diagramas de objetos y de secuencia

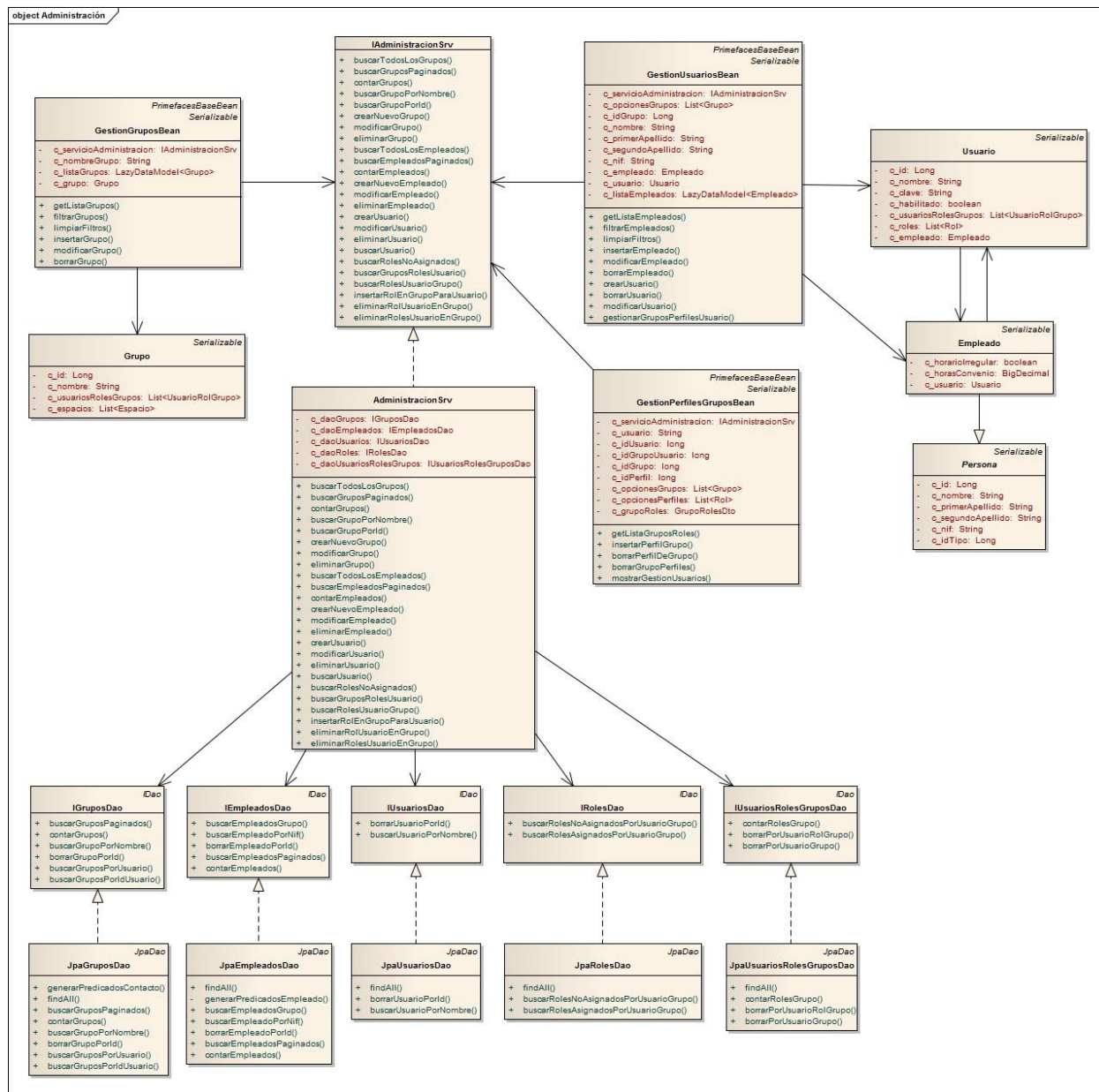
En este apartado se detallarán los diagramas de objetos y los diagramas de secuencia para cada una de las funcionalidades y operaciones que se pueden realizar sobre la aplicación.

En el apartado anterior se han definido los diagramas de clases estructurales, es decir, sin detallar las dependencias entre las clases de los distintos paquetes o agrupaciones que se definirán en la aplicación. Aquí se definirán los diagramas de objetos y los diagramas de secuencia para observar las dependencias existentes entre las clases de los distintos paquetes, y cómo se invocan los distintos métodos principales de cada una de ellas para llevar a cabo una determinada funcionalidad. No se entrará en detalle sobre las funcionalidades menores que se necesiten en la aplicación para dar apoyo a la capa de presentación, ni en la autenticación de usuario en el sistema, ya que para ello delegaremos en el framework de seguridad Spring Security.

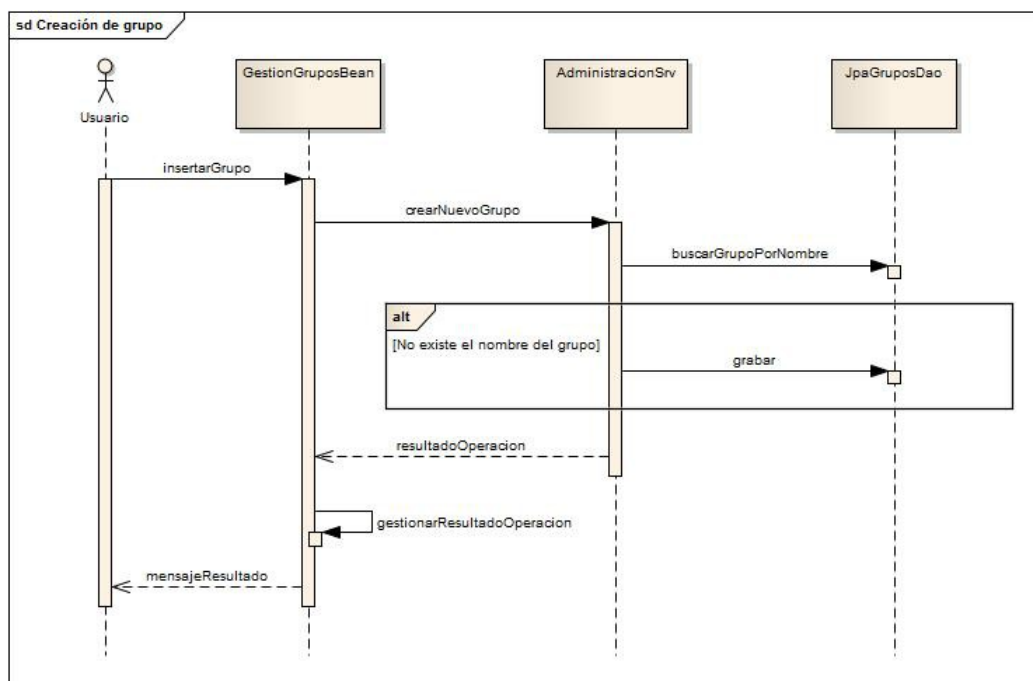
3.4.1 Funcionalidades de administración de la aplicación

Las funcionalidades de administración de la aplicación son todas las relacionadas con la gestión de grupos de trabajo, empleados y usuarios de la misma. Para implementar las funcionalidades de administración, se crearán tres clases de controladores, uno para la gestión de grupos, otro para la gestión de empleados y usuarios y el último para la gestión de roles de usuario en grupos. Además, estas clases utilizarán el servicio de negocio de administración, el cual a su vez, necesitará utilizar los objetos de acceso a datos de grupos, usuarios y usuarios roles y grupos.

El diagrama de objetos para las funcionalidades de administración será el siguiente:



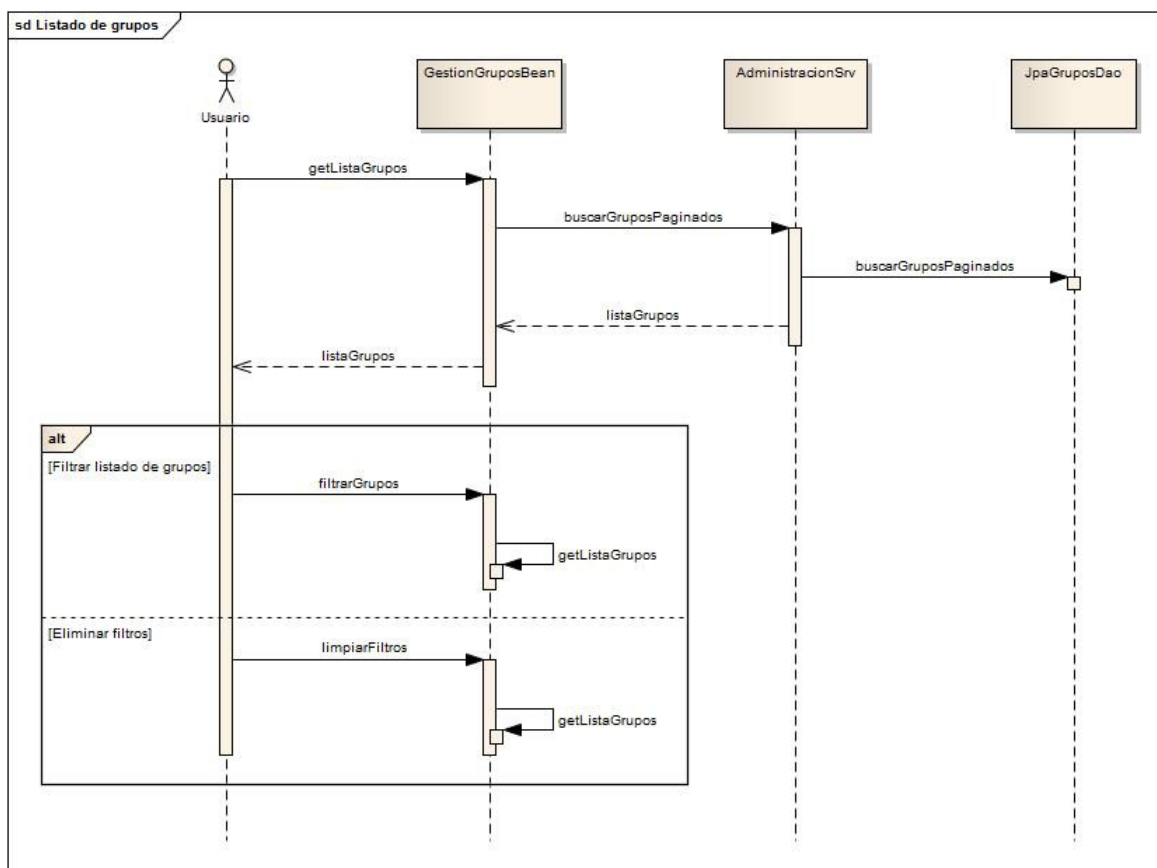
3.4.1.1 Diagrama de secuencia de la creación de grupo de trabajo



Para la creación de un nuevo grupo de trabajo, se habrá de realizar la siguiente secuencia:

1. El usuario invoca el método insertarGrupo del controlador GestionGruposBean, desde la página web de gestión de grupos.
2. Se realizan las validaciones necesarias sobre los datos introducidos por el usuario. Si no hay error se continúa y si hay error se muestran los mensajes correspondientes.
3. Se invoca al método de negocio crearNuevoGrupo del servicio AdministracionSrv.
4. Se valida que no exista un grupo con el mismo nombre mediante el método buscarGrupoPorNombre del DAO JpaGruposDao.
5. Si no existe ningún grupo con el nombre se graba el nuevo grupo con el método grabar del DAO.
6. Se devuelve el resultado de la operación.
7. Se trata el resultado de la operación.
8. Se muestra al usuario el mensaje de resultado de la operación.

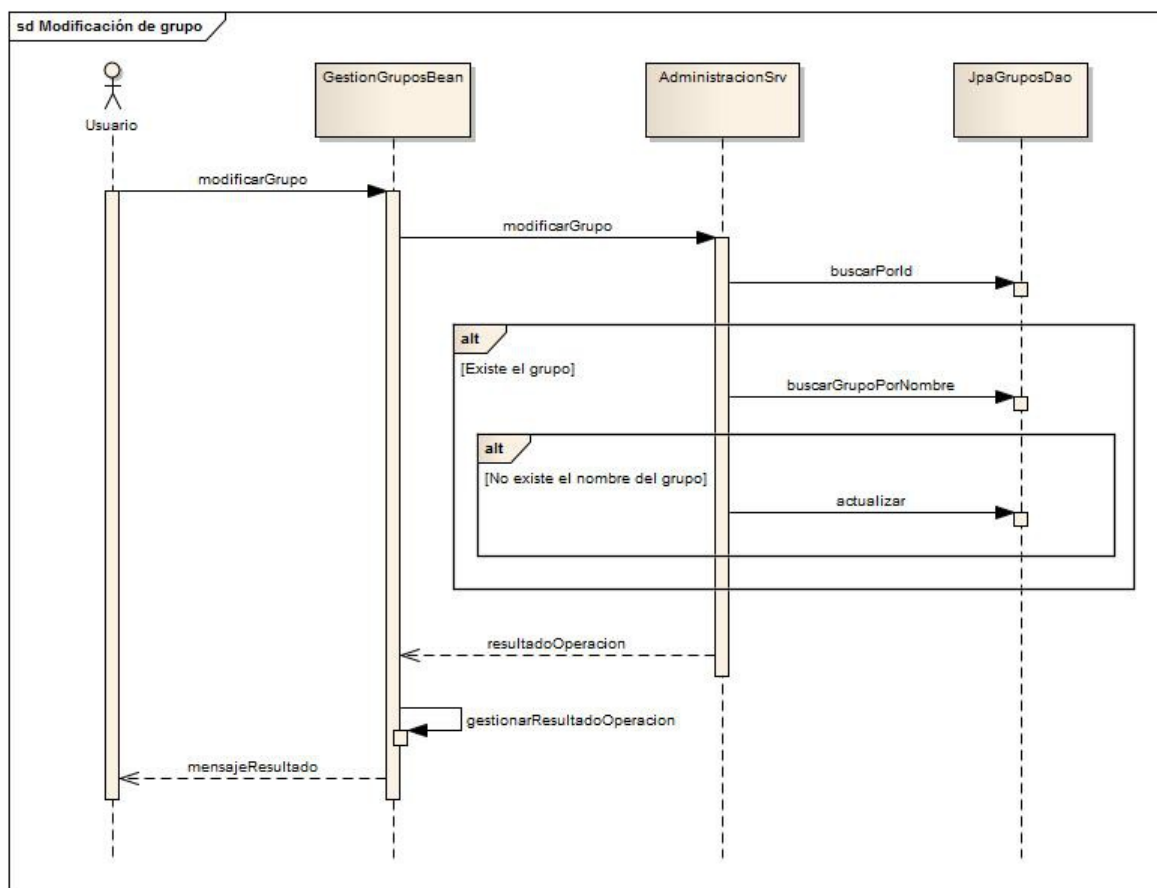
3.4.1.2 Diagrama de secuencia del listado de grupos de trabajo



Para la listar los grupos de trabajo, se realizará la siguiente secuencia:

1. El usuario invoca el método `getListaGrupos` del controlador `GestionGruposBean` desde la página web de listado de grupos.
2. Se consulta la primera página de grupos invocando el método `buscarGruposPaginados` del servicio `AdministracionSrv`.
3. Se consulta la página de base de datos mediante el método `buscarGruposPaginados` del DAO.
4. Se le muestra al usuario la página del listado o el mensaje de no existencia de grupos. Si hubiese más de una página se puede navegar entre ellas repitiendo la secuencia.
5. Alternativamente se puede filtrar el listado de grupos, con lo que se repetirá la secuencia pero realizando la búsqueda con los filtros introducidos.
6. También se podrán eliminar los filtros, con lo que se repetirá la secuencia tras limpiar los filtros.

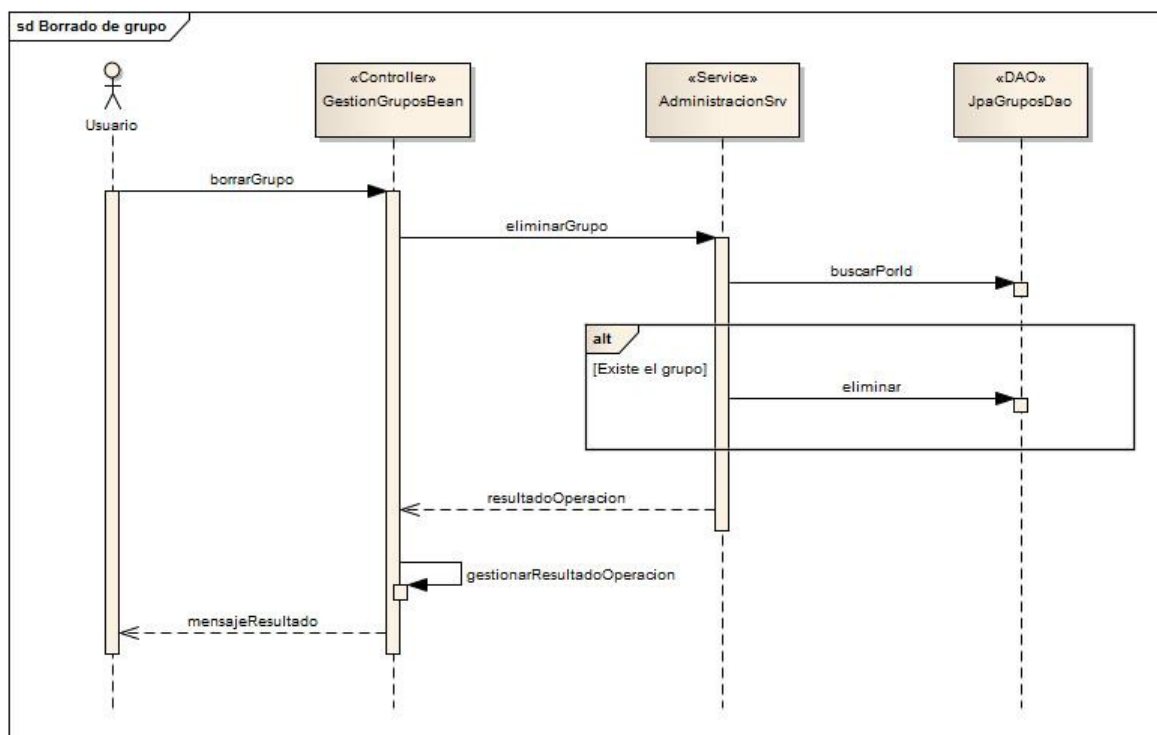
3.4.1.3 Diagrama de secuencia de la modificación de grupo de trabajo



Para la modificación de un grupo de trabajo, se realizará la siguiente secuencia:

1. El usuario invoca el método `modificarGrupo` del controlador `GestionGruposBean`, desde la página web del listado de grupos, seleccionando el grupo a modificar.
2. Se invoca al método `modificarGrupo` del servicio `AdministraciónSrv`.
3. Se comprueba que exista el grupo a modificar buscándolo por identificador con el método `buscarPorId` del DAO `JpaGruposDao`.
4. Si existe, se comprueba que no exista un grupo con el nuevo nombre a asignar.
5. Si no existe se actualiza el grupo.
6. Se devuelve el resultado de la operación.
7. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje correspondiente.

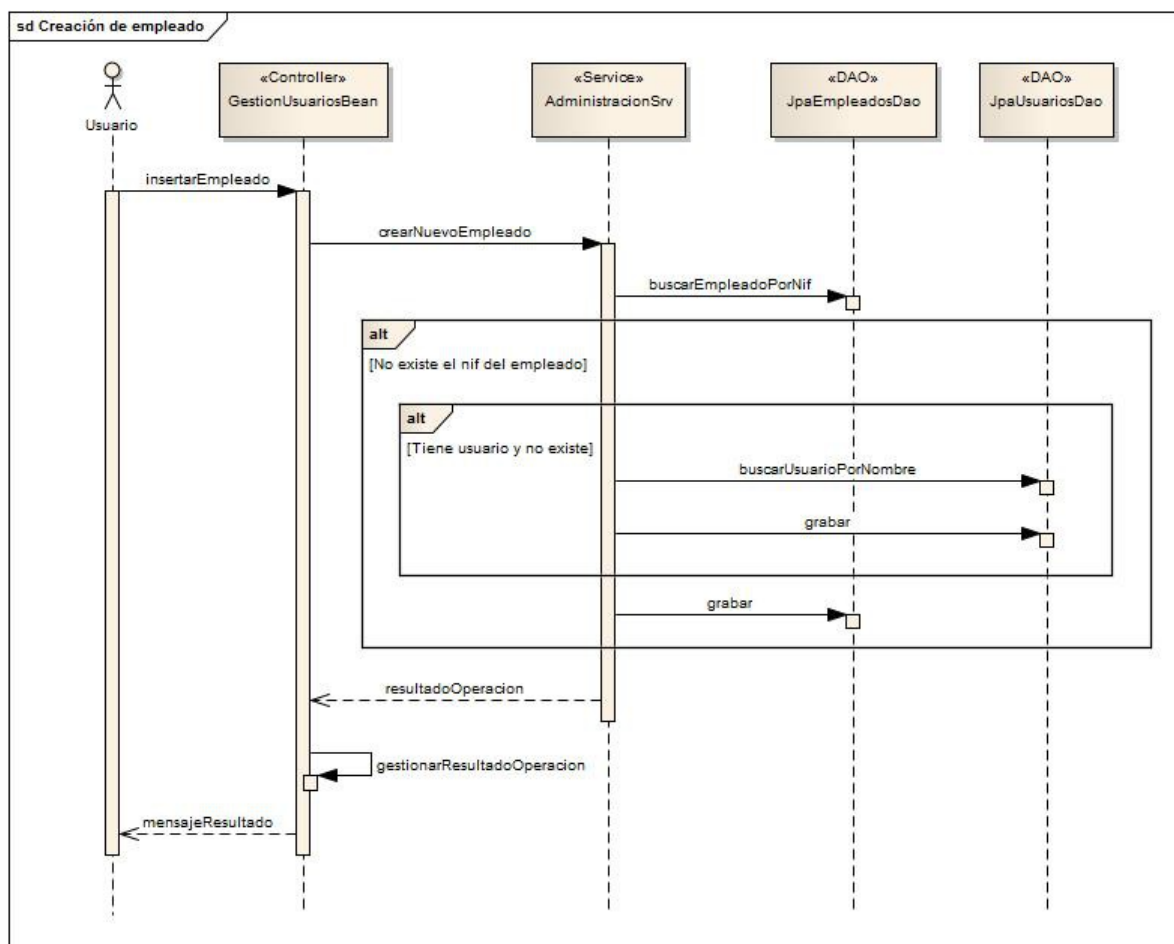
3.4.1.4 Diagrama de secuencia del borrado de grupo de trabajo



Para el borrado de un grupo de trabajo, se realizará la siguiente secuencia:

1. El usuario invoca el método `borrarGrupo` del controlador `GestionGruposBean`, desde la página web del listado de grupos, seleccionando el grupo a eliminar.
2. Se invoca al método `eliminarGrupo` del servicio `AdministraciónSrv`.
3. Se comprueba que exista el grupo a eliminar buscándolo por identificador con el método `buscarPorId` del DAO `JpaGruposDao`.
4. Si no tiene usuarios asignados se elimina, en caso contrario no se podrá eliminar.
5. Se devuelve el resultado de la operación.
6. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje correspondiente.

3.4.1.5 Diagrama de secuencia de la creación de empleado

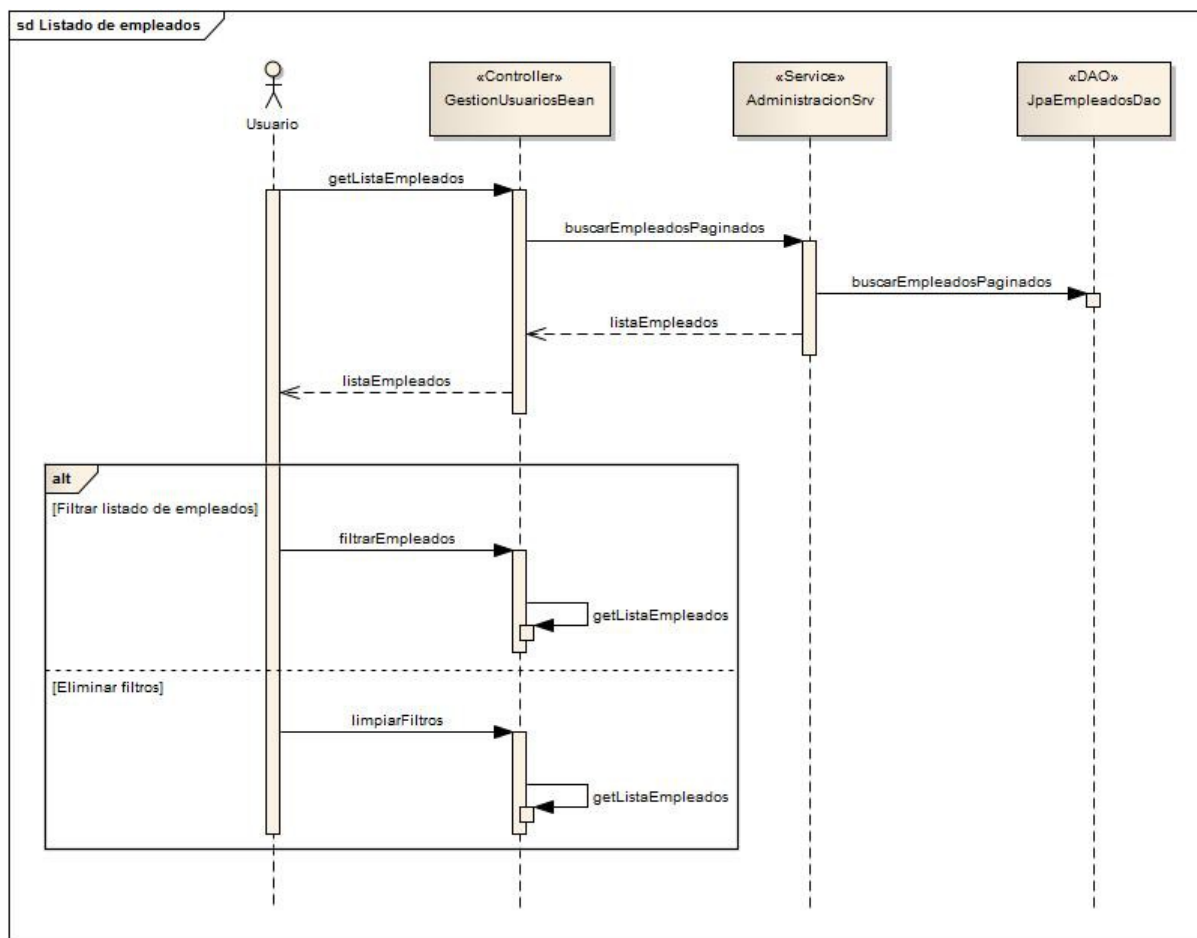


Para la creación de un nuevo empleado, se habrá de realizar la siguiente secuencia:

1. El usuario invoca el método `insertarEmpleado` del controlador `GestionUsuariosBean`, desde la página web de gestión de empleados.
2. Se realizan las validaciones necesarias sobre los datos introducidos por el usuario. Si no hay error se continúa y si hay error se muestran los mensajes correspondientes.
3. Se invoca al método de negocio `crearNuevoEmpleado` del servicio `AdministracionSrv`.
4. Se valida que no exista un empleado con el mismo NIF mediante el método `buscarGrupoPorNif` del DAO `JpaEmpleadosDao`.
5. Si no existe ningún empleado con el NIF se comprueba si se le ha asignado un usuario y si es así, se comprueba que no exista el mismo mediante el método `buscarUsuarioPorNombre` del `JpaUsuariosDao`.
6. Si no se ha asignado usuario o no existe se graba el empleado y/o el usuario.

7. Se devuelve el resultado de la operación.
8. Se trata el resultado de la operación y se muestra al usuario el mensaje de resultado de la misma.

3.4.1.6 Diagrama de secuencia del listado de empleados



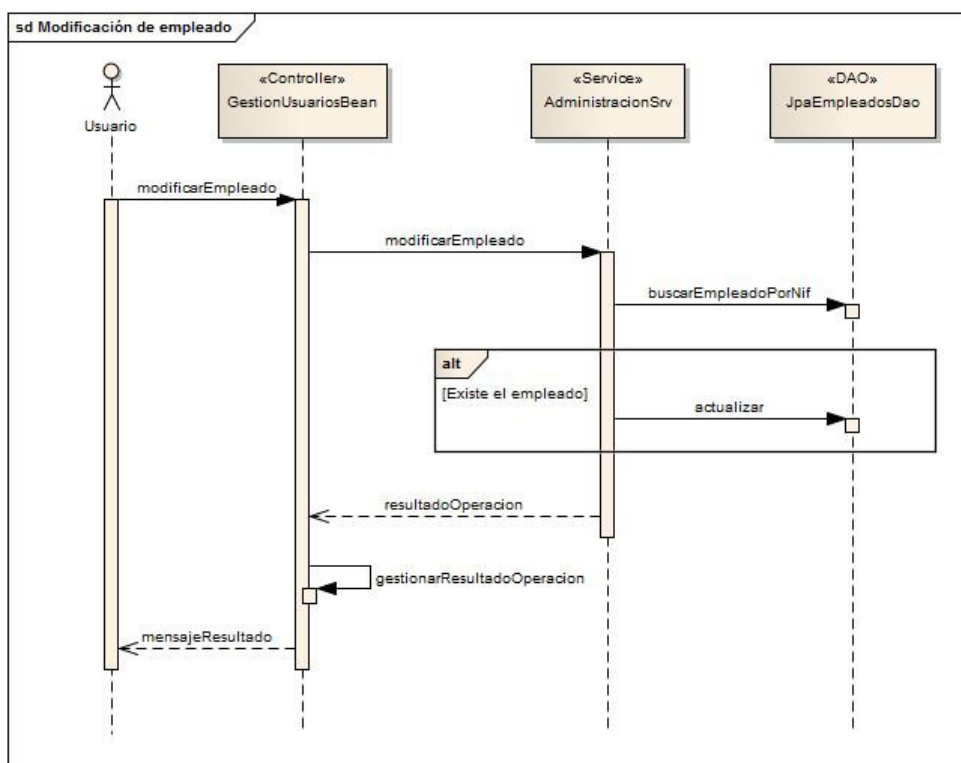
Para la listar los empleados y usuarios, se realizará la siguiente secuencia:

1. El usuario invoca el método `getListaEmpleados` del controlador `GestionUsuariosBean` desde la página web de listado de empleados y usuarios.
2. Se consulta la primera página de empleados invocando el método `buscarEmpleadosPaginados` del servicio `AdministracionSrv`.
3. Se consulta la página de base de datos mediante el método `buscarEmpleadosPaginados` del `JpaEmpleadosDao`.
4. Se le muestra al usuario la página del listado o el mensaje de no existencia de empleados según corresponda. Si hubiese más de una página se puede navegar entre ellas repitiendo la

secuencia.

5. Alternativamente se puede filtrar el listado de empleados, con lo que se repetirá la secuencia pero realizando la búsqueda con los filtros introducidos.
6. También se podrán eliminar los filtros, con lo que se repetirá la secuencia tras limpiar los filtros.

3.4.1.7 Diagrama de secuencia de la modificación de empleado

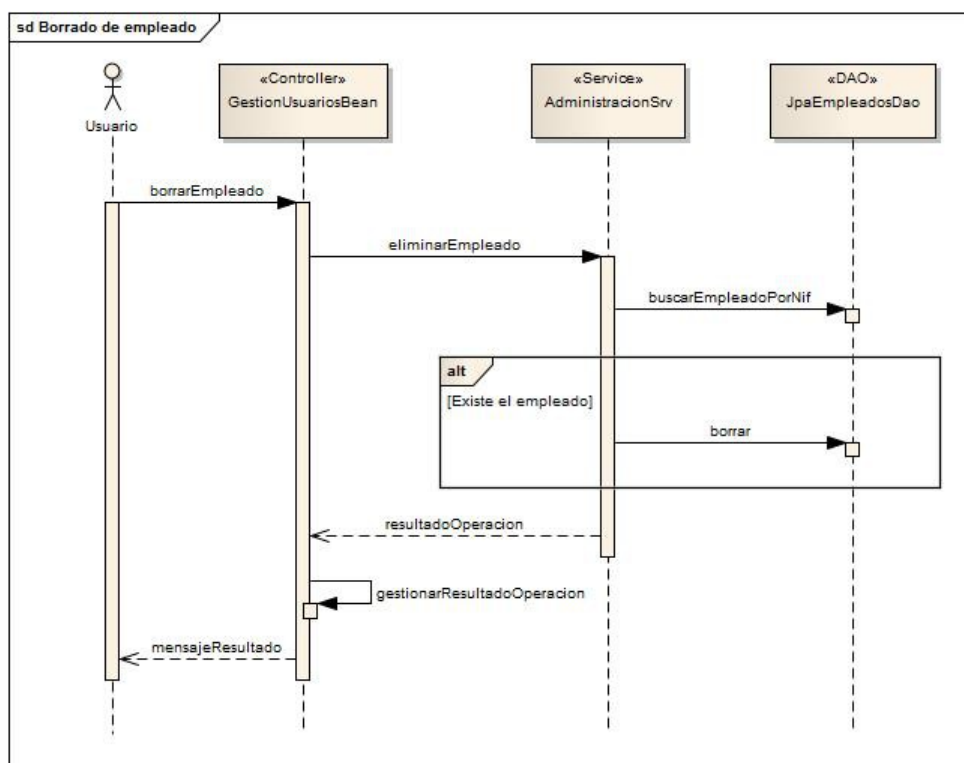


Para la modificación de un empleado, se realizará la siguiente secuencia:

1. El usuario invoca el método `modificarEmpleado` del controlador `GestionUsuariosBean`, desde la página web del listado de empleados, seleccionando el empleado a modificar.
2. Se realizan las validaciones de los nuevos datos introducidos por el usuario. Si son correctas se continuará con la modificación, si no, se devolverá el mensaje de error.
3. Se invoca al método `modificarEmpleado` del servicio `AdministraciónSrv`.
4. Se comprueba que exista el empleado a modificar buscándolo por identificador con el método `buscarPorId` del DAO `JpaEmpleadosDao`.
5. Si existe, se comprueba si se ha modificado el NIF, en cuyo caso habrá que comprobar que no exista ya un empleado con el NIF introducido.

6. Si existe el empleado y no tiene el mismo NIF que otro existente se modifica.
7. Se devuelve el resultado de la operación.
8. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje correspondiente.

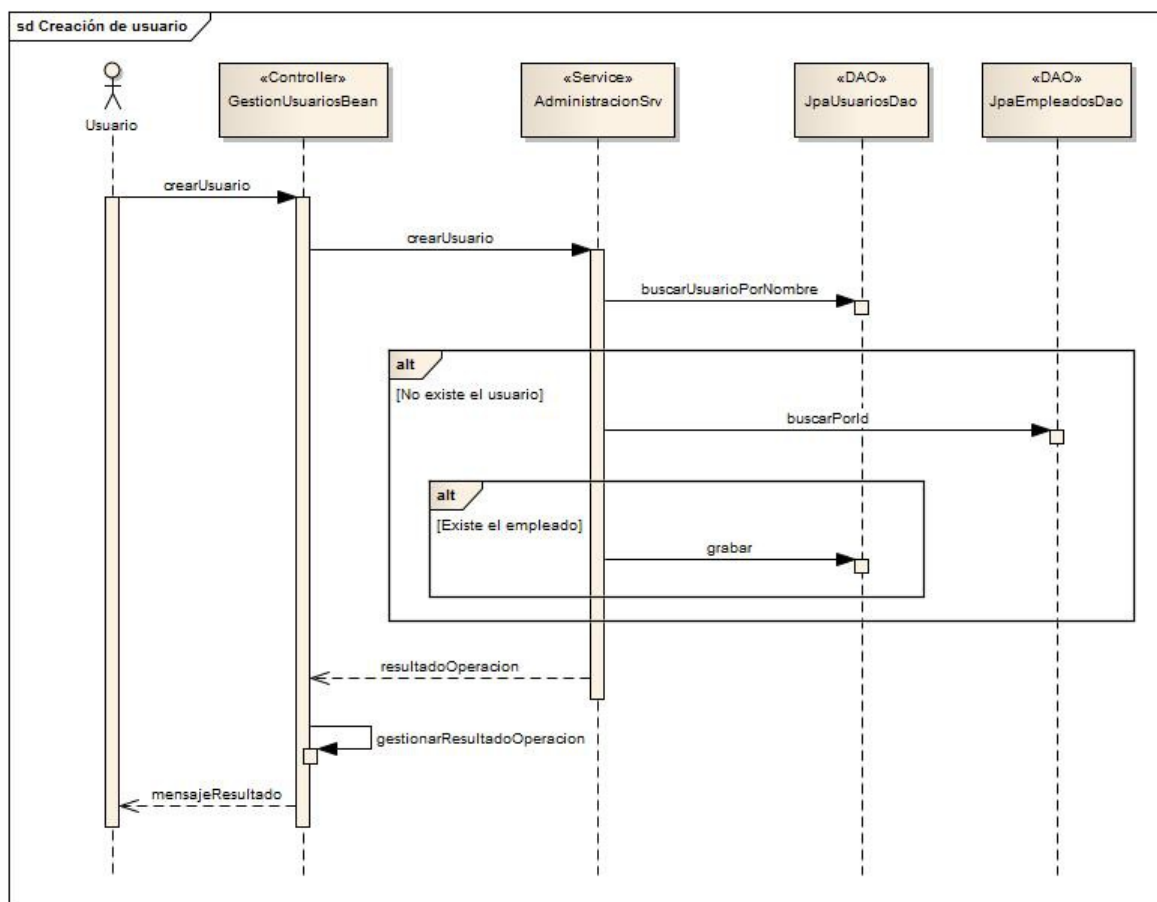
3.4.1.8 Diagrama de secuencia del borrado de empleado



Para el borrado de un empleado, se realizará la siguiente secuencia:

1. El usuario invoca el método borrarEmpleado del controlador GestionUsuariosBean, desde la página web del listado de empleados, seleccionando el empleado a eliminar.
2. Se invoca al método eliminarEmpleado del servicio AdministraciónSrv.
3. Se comprueba que exista el empleado a eliminar buscándolo por identificador con el método buscarPorId del DAO JpaEmpleadosDao.
4. Se elimina el empleado y el usuario asignado al mismo si lo tuviese.
5. Se devuelve el resultado de la operación.
6. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje correspondiente.

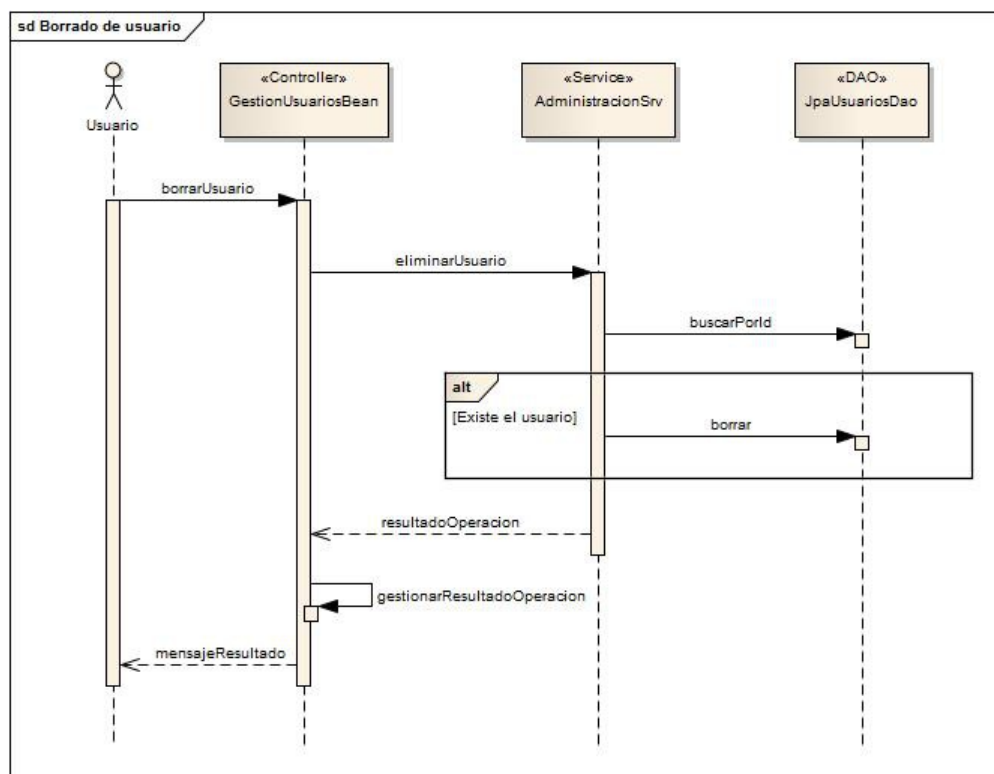
3.4.1.9 Diagrama de secuencia de la creación de usuario



Para la creación de un nuevo empleado, se habrá de realizar la siguiente secuencia:

1. El usuario invoca el método `crearUsuario` del controlador `GestionUsuariosBean`, desde la página web de listado de empleados, seleccionado el empleado para el cual crear el usuario.
2. Se realizan las validaciones necesarias sobre los datos introducidos por el usuario. Si no hay error se continúa y si hay error se muestran los mensajes correspondientes.
3. Se invoca al método de negocio `crearUsuario` del servicio `AdministracionSrv`.
4. Se valida que no exista un usuario con el mismo nombre mediante el método `buscarUsuarioPorNombre` del DAO `JpaUsuariosDao`.
5. Si no existe ningún usuario con el mismo nombre se obtiene el empleado mediante su identificador, y si existe se crea el usuario y se le asigna.
6. Se devuelve el resultado de la operación.
7. Se trata el resultado de la operación y se muestra al usuario el mensaje de resultado de la misma.

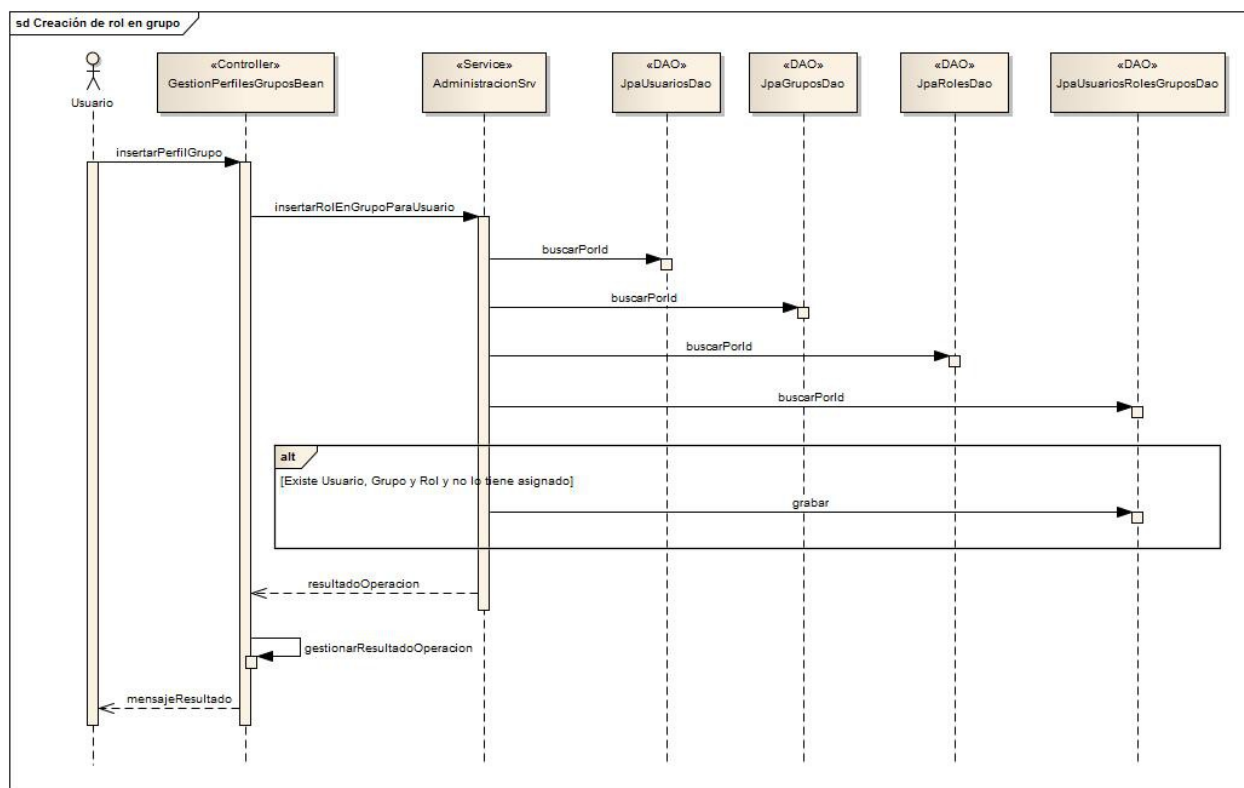
3.4.1.10 Diagrama de secuencia del borrado de usuario



Para el borrado de un usuario, se realizará la siguiente secuencia:

1. El usuario invoca el método `borrarUsuario` del controlador `GestionUsuariosBean`, desde la página web del listado de empleados, seleccionando el usuario del empleado a eliminar.
2. Se invoca al método `eliminarUsuario` del servicio `AdministraciónSrv`.
3. Se comprueba que exista el usuario a eliminar buscándolo por identificador con el método `buscarPorId` del DAO `JpaUsuariosDao`.
4. Se elimina el usuario.
5. Se devuelve el resultado de la operación.
6. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje correspondiente.

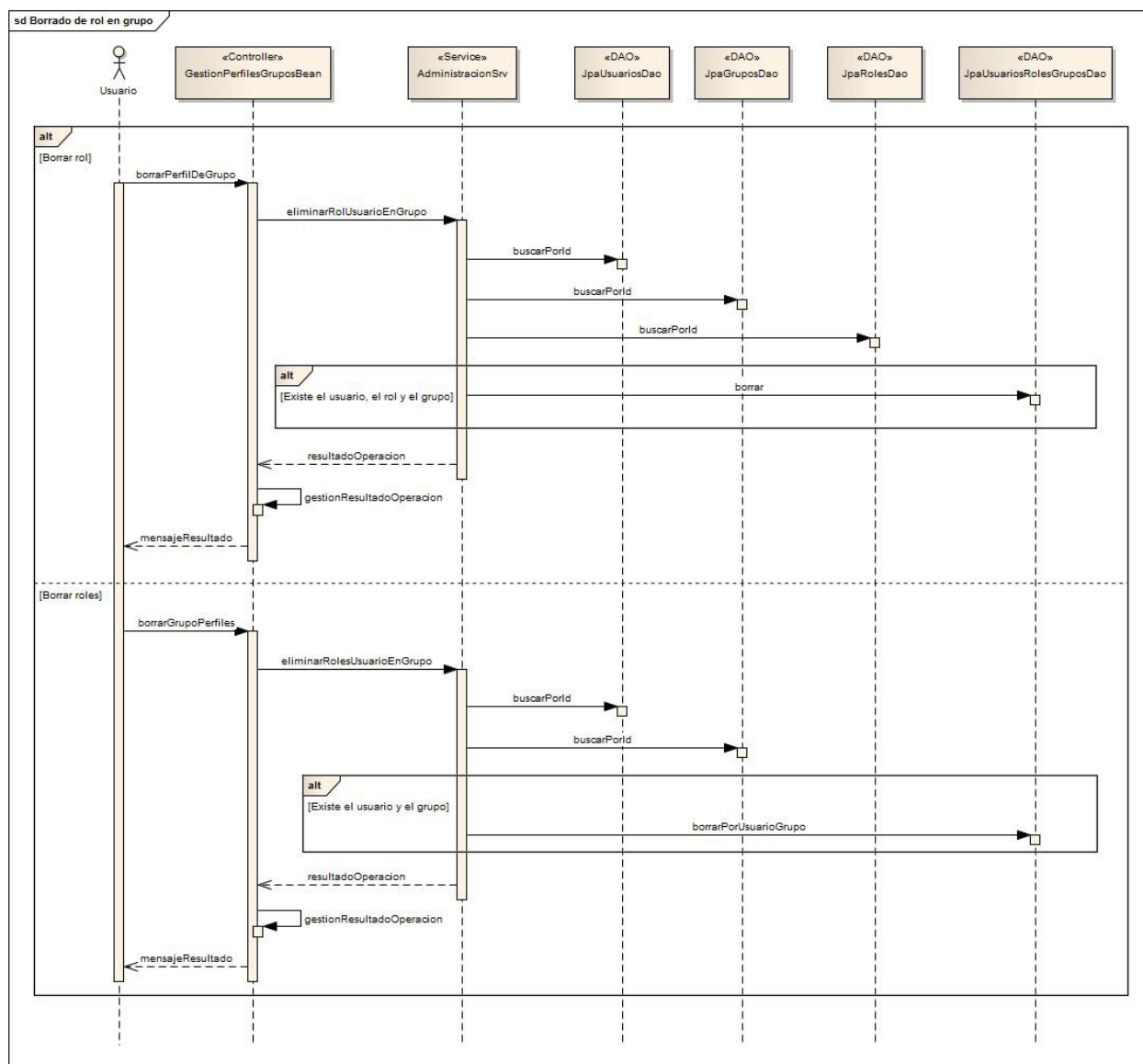
3.4.1.11 Diagrama de secuencia de la inserción de rol de usuario en grupo



Para la creación de un nuevo perfil en un grupo para un usuario, se habrá de realizar la siguiente secuencia:

1. El usuario invoca el método `insertarRolEnGrupoParaUsuario` del controlador `GestionPerfilesGruposBean`, desde la página web de gestión de perfiles y grupos de usuarios.
2. Se selecciona el grupo y el perfil a insertar para el usuario.
3. Se invoca al método de negocio `crearRolEnGrupoParaUsuario` del servicio `AdministracionSrv`.
4. Se valida que existan el usuario, el grupo y el rol seleccionados mediante los métodos de búsqueda por Id de sus respectivos DAO, `JpaUsuariosDao`, `JpaGruposDao` y `JpaRolesDao`.
5. Si existen el usuario, el perfil y el grupo y no los tiene ya asignados, se insertan el perfil y el grupo para el usuario mediante el método `grabar` del `JpaUsuariosRolesGruposDao`.
6. Se devuelve el resultado de la operación.
7. Se trata el resultado de la operación y se muestra al usuario el mensaje de resultado de la misma.

3.4.1.12 Diagrama de secuencia del borrado de rol de usuario en grupo



Para el borrado de un rol de usuario en un grupo, se realizará la siguiente secuencia:

1. El usuario invoca el método `borrarPerfilDeGrupo` del controlador `GestorPerfilesGruposBean`, desde la página web del listado de perfiles y grupos de usuario y seleccionando el grupo y el perfil del usuario en el mismo a eliminar.
2. Se invoca al método `eliminarRolUsuarioEnGrupo` del servicio `AdministraciónSrv`.
3. Se comprueba que exista el perfil en el grupo para el usuario.
4. Se elimina el perfil en el grupo.
5. Se devuelve el resultado de la operación y se le muestra al usuario el mensaje de resultado de la

misma.

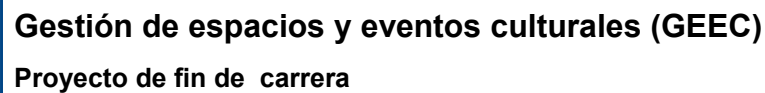
Para el borrado de todos los roles de un usuario en un grupo, se realizará la siguiente secuencia:

1. El usuario invoca el método `borrarGrupoPerfiles` del controlador `GestionPerfilesGruposBean`, desde la página web del listado de perfiles y grupos de usuario y seleccionando el grupo para el cual eliminar todos los perfiles del usuario.
2. Se invoca al método `eliminarRolesUsuarioEnGrupo` del servicio `AdministraciónSrv`.
3. Se comprueba que el usuario tenga perfiles en el grupo y se eliminan todos.
4. Se devuelve el resultado de la operación y se le muestra al usuario el mensaje de resultado de la misma.

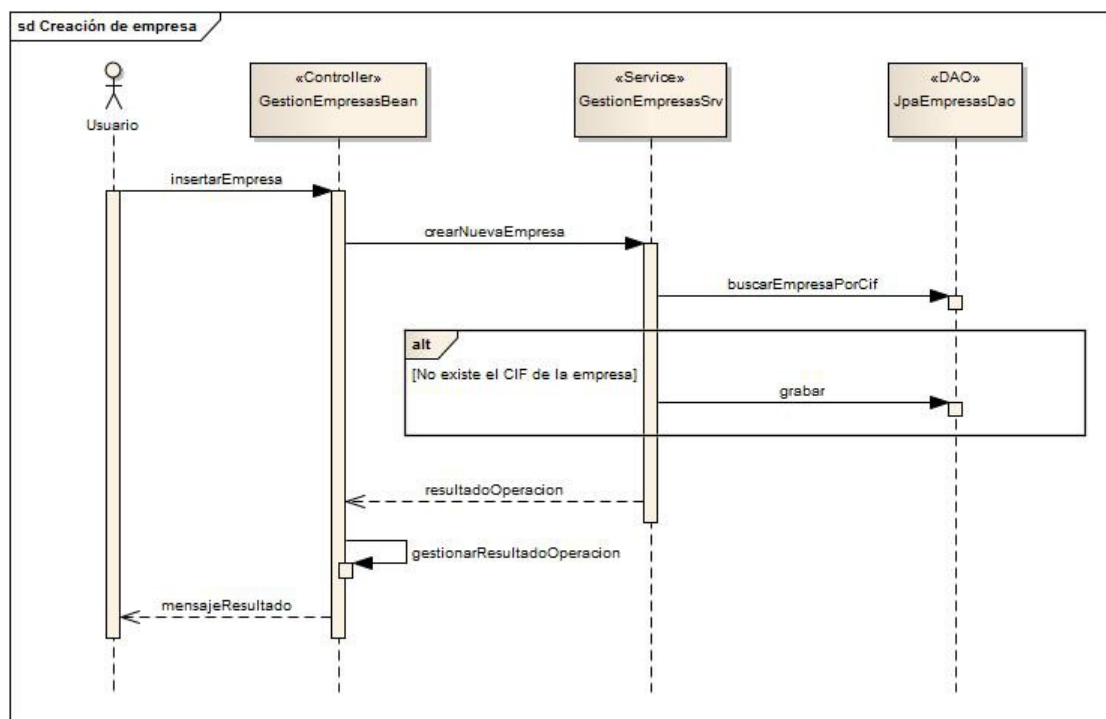
3.4.2 Funcionalidades de gestión de empresas y contactos

Para implementar las funcionalidades de gestión de empresas y contactos, se crearán dos controladores, uno para la gestión de empresas y otro para la gestión de contactos, con todas las operaciones que se puedan realizar sobre las mismas. Además, estas clases utilizarán el servicio de negocio de gestión de empresas, el cual a su vez, necesitará utilizar los objetos de acceso a datos de empresas, actividades, comunidades autónomas, provincias, municipios, contactos, eventos e histórico de eventos.

El diagrama de objetos para las funcionalidades de gestión de empresas y contactos será el siguiente:



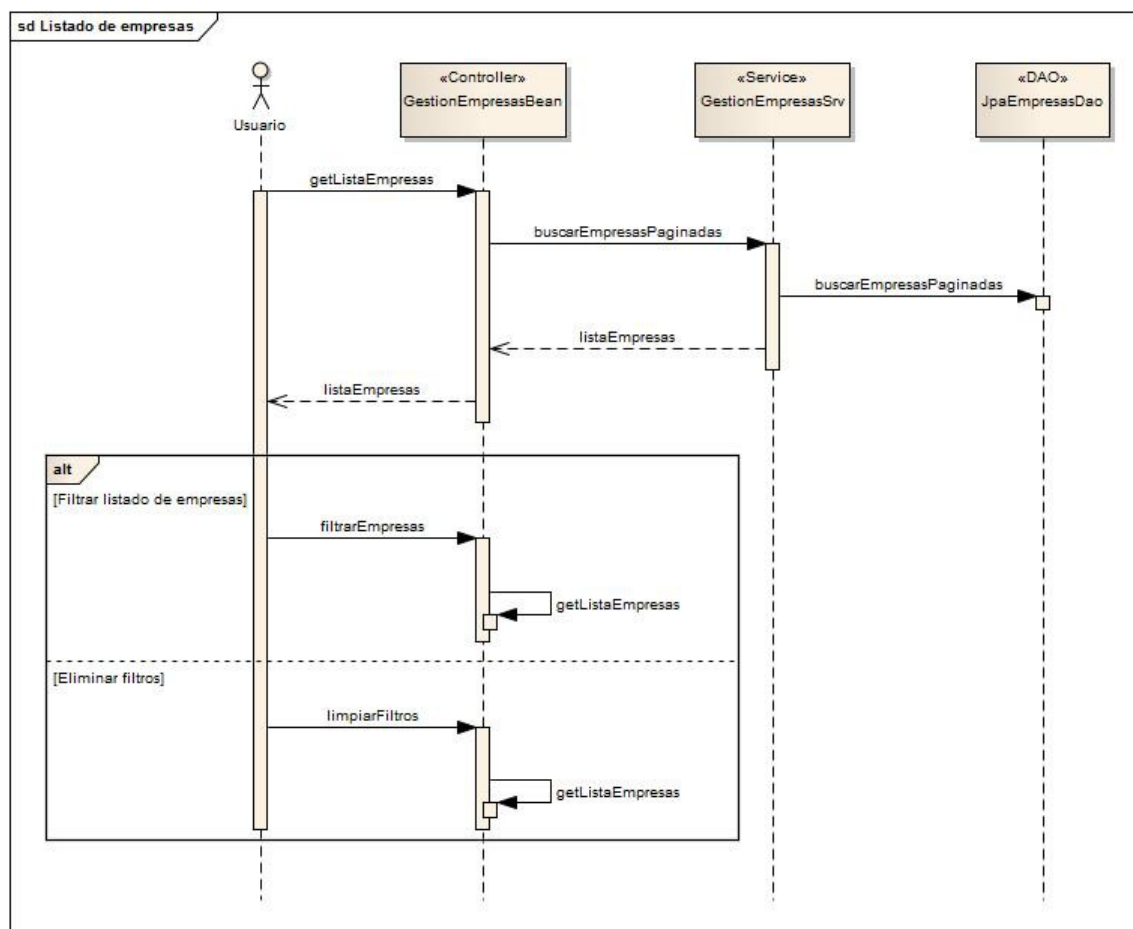
3.4.2.1 Diagrama de secuencia de la creación empresa



Para la creación de una nueva empresa, se habrá de realizar la siguiente secuencia:

1. El usuario invoca el método `insertarEmpresa` del controlador `GestionEmpresasBean`, desde la página web de gestión de empresas.
2. Se realizan las validaciones necesarias sobre los datos introducidos por el usuario. Si no hay error se continúa y si hay error se muestran los mensajes correspondientes.
3. Se invoca al método de negocio `crearNuevaEmpresa` del servicio `GestionEmpresasSrv`.
4. Se valida que no exista una empresa con el mismo CIF mediante el método `buscarEmpresaPorCif` del DAO `JpaEmpresasDao`.
5. Si no existe ninguna empresa con el CIF se graba la nueva empresa con el método `grabar` del DAO `JpaEmpresasDao`.
6. Se devuelve el resultado de la operación.
7. Se trata el resultado de la operación y se muestra al usuario el mensaje de resultado de la misma.

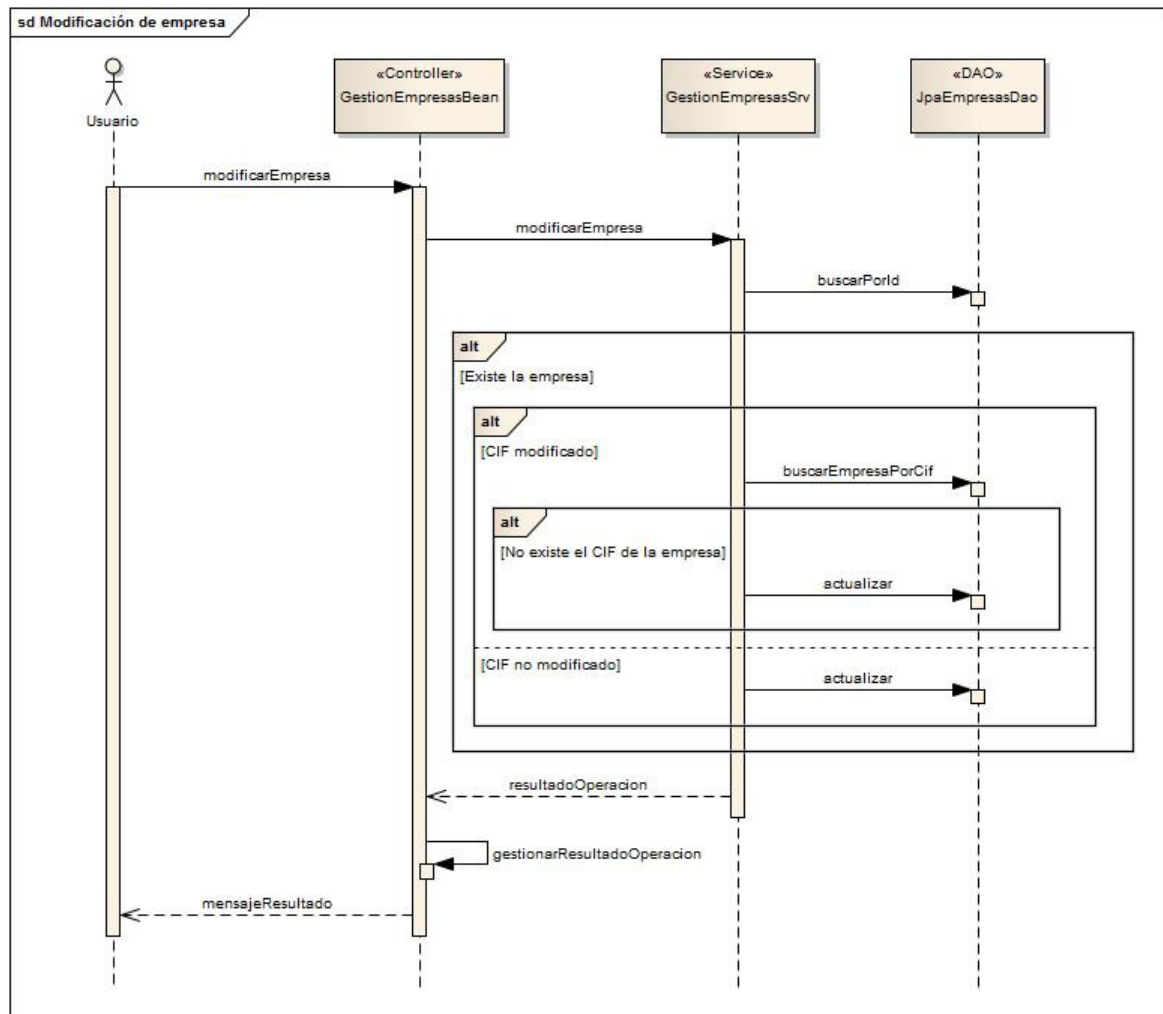
3.4.2.2 Diagrama de secuencia del listado de empresas



Para la listar las empresas, se realizará la siguiente secuencia:

1. El usuario invoca el método `getListaEmpresas` del controlador `GestionEmpresasBean` desde la página web de listado de empresas.
2. Se consulta la primera página de empresas invocando el método `buscarEmpresasPaginadas` del servicio `GestionEmpresasSrv`.
3. Se consulta la página de base de datos mediante el método `buscarEmpresasPaginadas` del DAO.
4. Se le muestra al usuario la página del listado o el mensaje de no existencia de empresas según corresponda. Si hubiese más de una página se puede navegar entre ellas repitiendo la secuencia.
5. Alternativamente se puede filtrar el listado de empresas, con lo que se repetirá la secuencia pero realizando la búsqueda con los filtros introducidos.
6. También se podrán eliminar los filtros, con lo que se repetirá la secuencia tras limpiar los filtros.

3.4.2.3 Diagrama de secuencia de la modificación de empresa

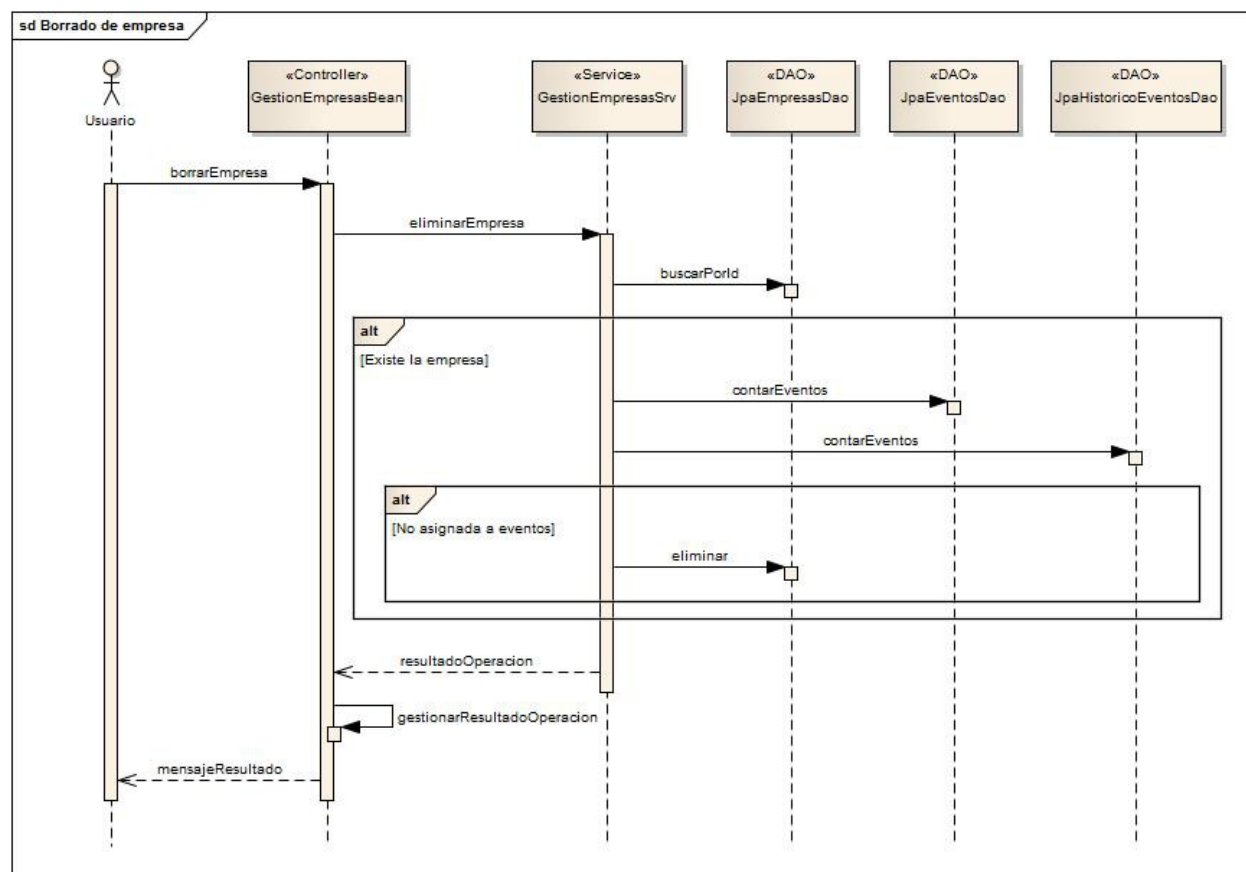


Para la modificación de una empresa, se realizará la siguiente secuencia:

1. El usuario invoca el método `modificarEmpresa` del controlador `GestionEmpresasBean`, desde la página web del listado de empresas, seleccionando la empresa a modificar.
2. Se invoca al método `modificarEmpresa` del servicio `GestionEmpresasSrv`.
3. Se comprueba que exista la empresa a modificar buscándola por identificador con el método `buscarPorId` del DAO `JpaEmpresasDao`.
4. Si existe, se comprueba si se ha modificado el CIF y si es así, que no exista una empresa con el mismo CIF a asignar.
5. Si no se ha modificado o no existe el nuevo CIF se actualiza la empresa.
6. Se devuelve el resultado de la operación.

7. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje correspondiente.

3.4.2.4 Diagrama de secuencia del borrado de empresa

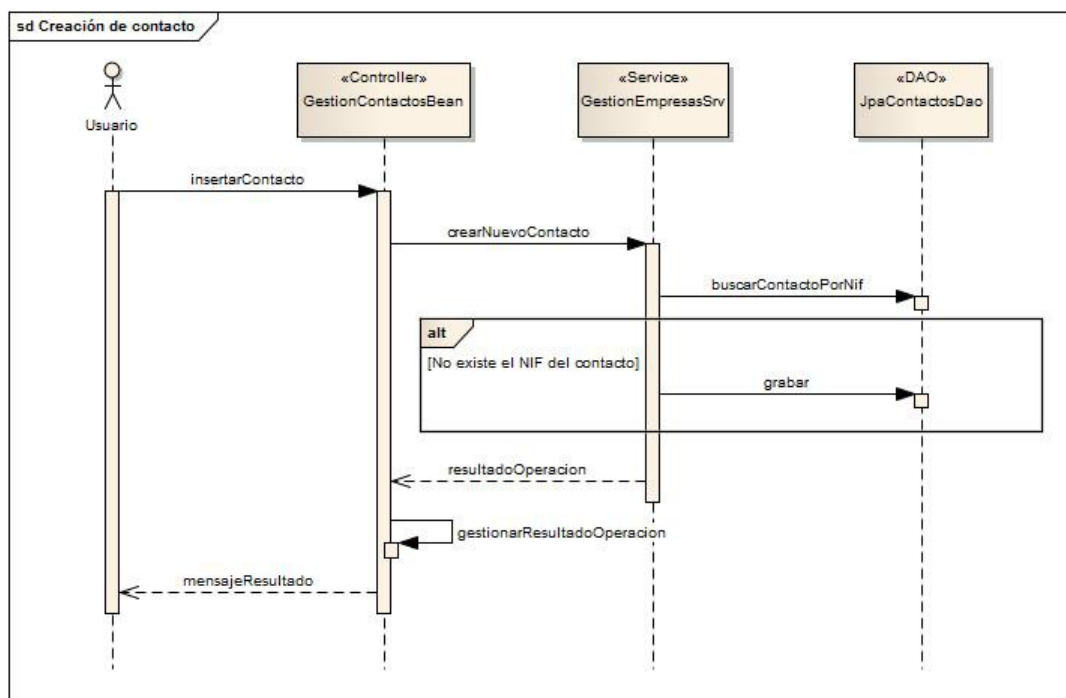


Para el borrado de una empresa, se realizará la siguiente secuencia:

1. El usuario invoca el método `borrarEmpresa` del controlador `GestionEmpresasBean`, desde la página web del listado de empresas, seleccionando la empresa a eliminar.
2. Se invoca al método `eliminarEmpresa` del servicio `GestionEmpresasSrv`.
3. Se comprueba que exista la empresa a eliminar buscándola por identificador con el método `buscarPorId` del DAO `JpaEmpresasDao`.
4. Si existe la empresa, se comprueba que la empresa no esté asignada a ningún evento o eventos que se encuentren en el histórico utilizando los correspondientes DAO `JpaEventosDao` y `JpaHistoricoEventosDao`, en cuyo caso se elimina la empresa.
5. Se devuelve el resultado de la operación al controlador.
6. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje

correspondiente al resultado de la misma, informando que se ha realizado correctamente o los problemas que se hayan encontrado.

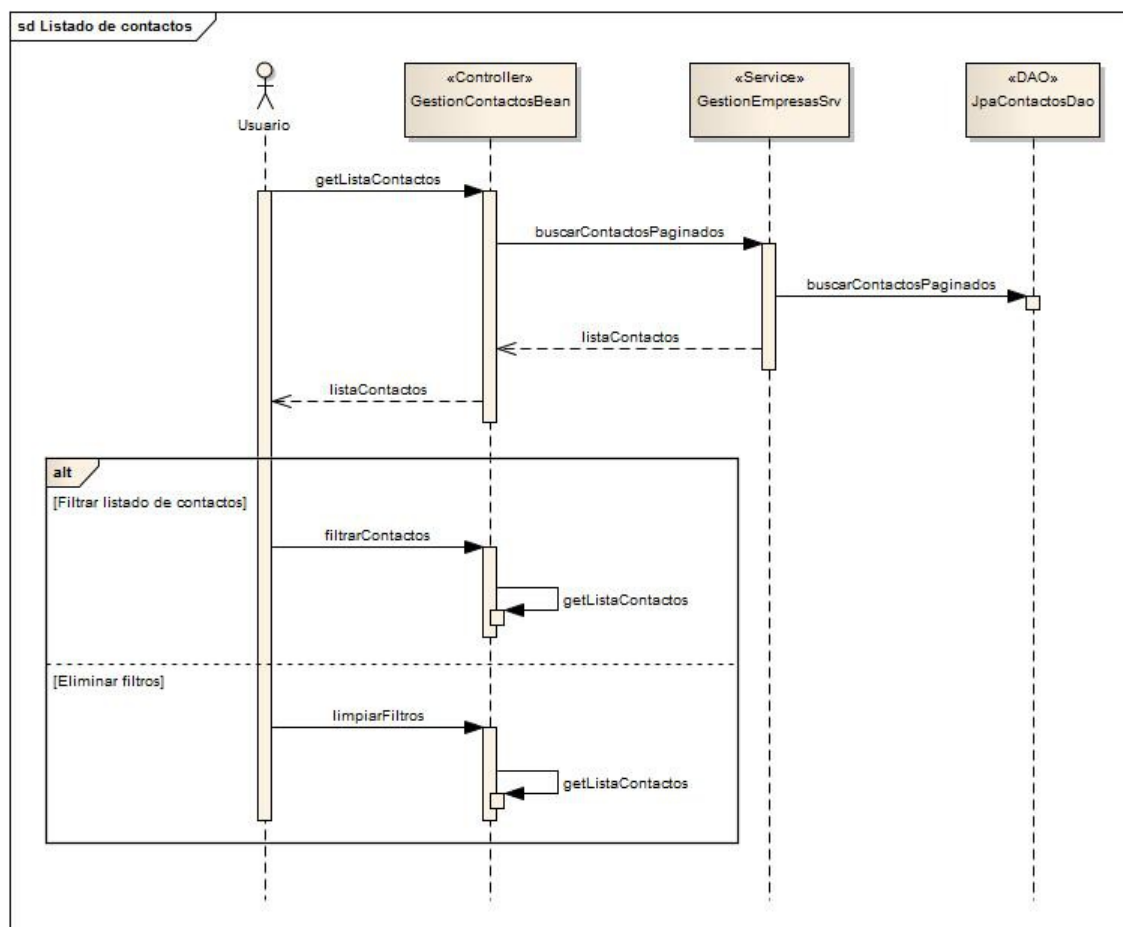
3.4.2.5 Diagrama de secuencia de la creación de contacto



Para la creación de un nuevo contacto, se habrá de realizar la siguiente secuencia:

1. El usuario invoca el método `insertarContacto` del controlador `GestionContactosBean`, desde la página web de gestión de contactos.
2. Se realizan las validaciones necesarias sobre los datos introducidos por el usuario. Si no hay error se continúa y si hay error se muestran los mensajes correspondientes.
3. Se invoca al método de negocio `crearNuevoContacto` del servicio `GestionContactosSrv`.
4. Se valida que no exista un contacto con el mismo NIF mediante el método `buscarContactoPorNif` del DAO `JpaContactosDao`.
5. Si no existe ningún contacto con el NIF se graba el nuevo contacto con el método `grabar` del DAO `JpaContactosDao`.
6. Se devuelve el resultado de la operación.
7. Se trata el resultado de la operación y se muestra al usuario el mensaje de resultado de la misma.

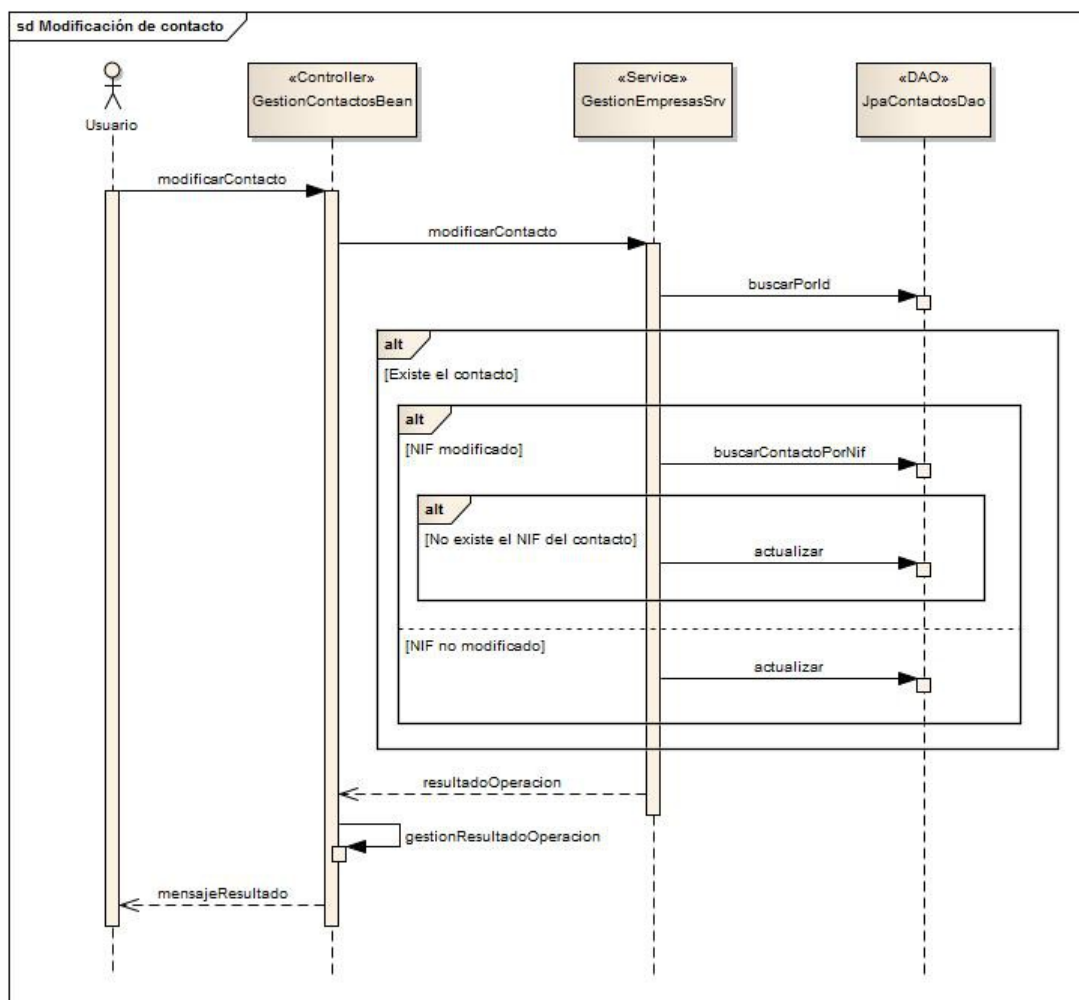
3.4.2.6 Diagrama de secuencia del listado de contactos



Para la listar los contactos de contactos, se realizará la siguiente secuencia:

1. El usuario invoca el método `getListaContactos` del controlador `GestionContactosBean` desde la página web de listado de contactos.
2. Se consulta la primera página de contactos invocando el método `buscarContactosPaginados` del servicio `GestionEmpresasSrv`.
3. Se consulta la página de base de datos mediante el método `buscarContactosPaginados` del DAO `JpaContactosDao`.
4. Se le muestra al usuario la página del listado o el mensaje de no existencia de contactos. Si hubiese más de una página se puede navegar entre ellas repitiendo la secuencia.
5. Alternativamente se puede filtrar el listado de contactos, con lo que se repetirá la secuencia pero realizando la búsqueda con los filtros introducidos.
6. También se podrán eliminar los filtros, con lo que se repetirá la secuencia tras limpiar los filtros.

3.4.2.7 Diagrama de secuencia de la modificación de contacto

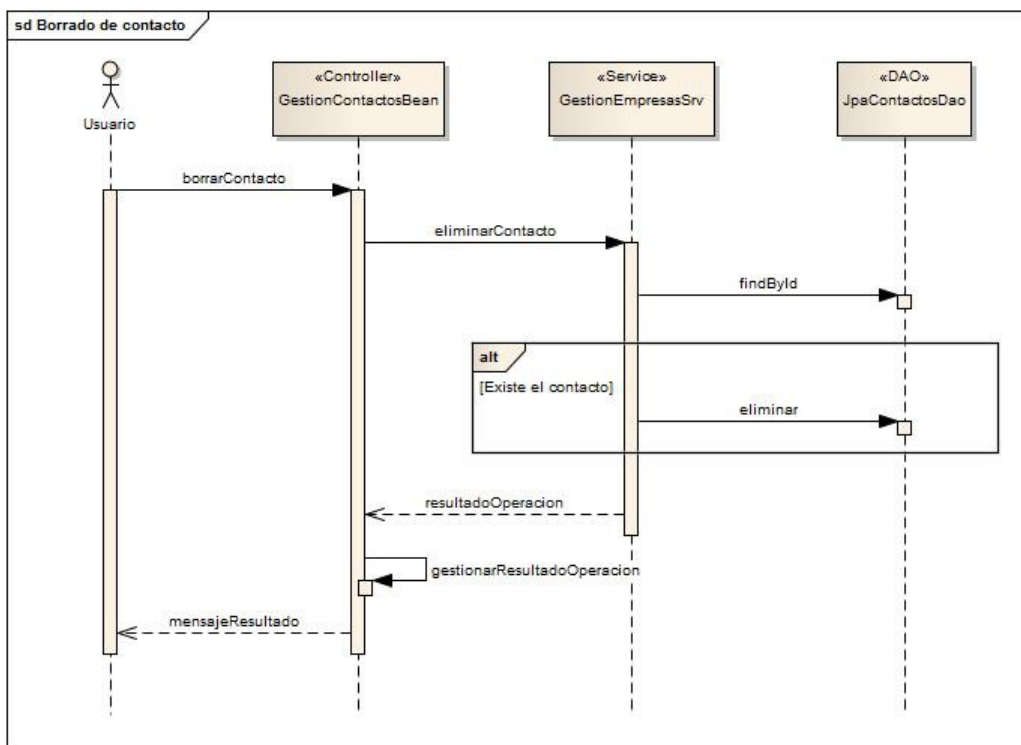


Para la modificación de un contacto, se realizará la siguiente secuencia:

1. El usuario invoca el método `modificarContacto` del controlador `GestionContactosBean`, desde la página web del listado de contactos, seleccionando el contacto a modificar.
2. Se invoca al método `modificarContacto` del servicio `GestionEmpresasSrv`.
3. Se comprueba que exista el contacto a modificar buscándolo por identificador con el método `buscarPorId` del DAO `JpaContactosDao`.
4. Si existe, se comprueba si se ha modificado el NIF del mismo, en cuyo caso habrá que comprobar que no exista un contacto con el nuevo NIF. Si no se ha modificado el NIF o no existe el nuevo NIF se actualiza el contacto.
5. Se devuelve el resultado de la operación.
6. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje

correspondiente.

3.4.2.8 Diagrama de secuencia del borrado de contacto



Para el borrado de un contacto, se realizará la siguiente secuencia:

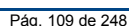
1. El usuario invoca el método `borrarContacto` del controlador `GestionContactosBean`, desde la página web del listado de contactos, seleccionando el contacto a eliminar.
2. Se invoca al método `eliminarContacto` del servicio `GestionContactosSrv`.
3. Se comprueba que exista el contacto a eliminar buscándolo por identificador con el método `buscarPorId` del DAO `JpaContactosDao`.
4. Si existe el contacto se elimina.
5. Se devuelve el resultado de la operación.
6. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje correspondiente.

3.4.3 Funcionalidades de gestión de espacios y eventos

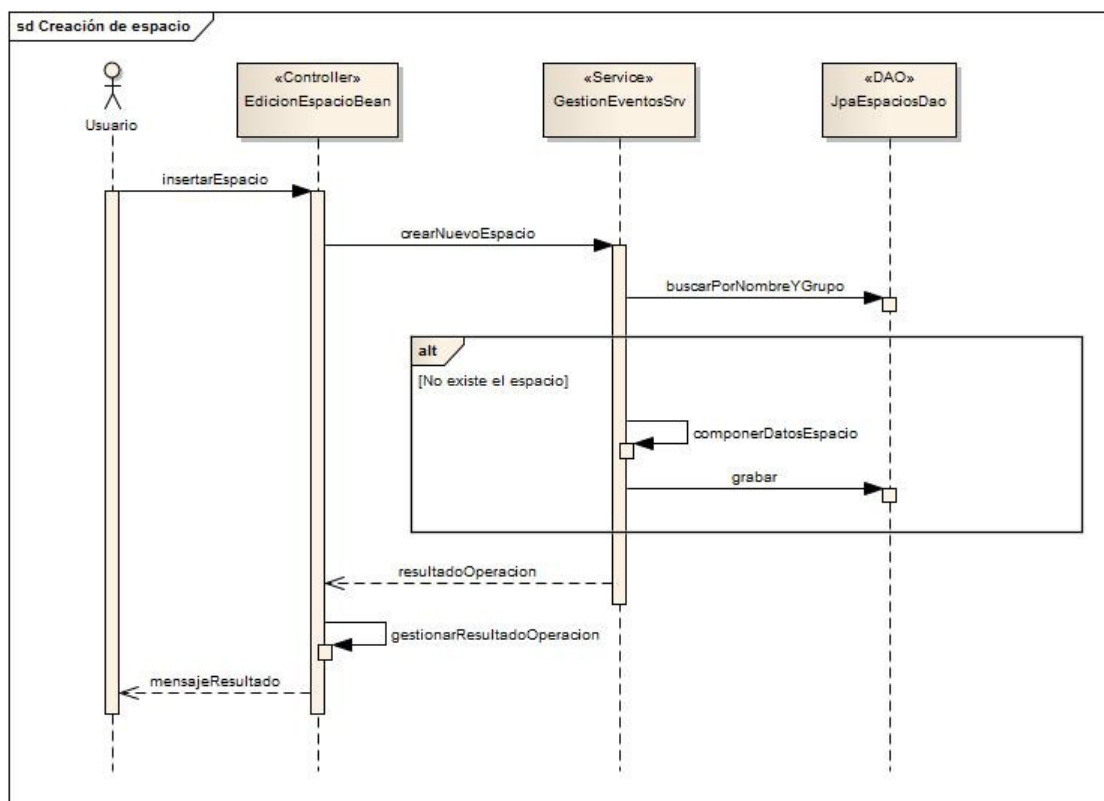
Para desarrollar las funcionalidades de gestión de espacios y eventos, se utilizarán cinco controladores,



El diagrama de objetos para las funcionalidades de gestión de espacios y eventos culturales será el siguiente:



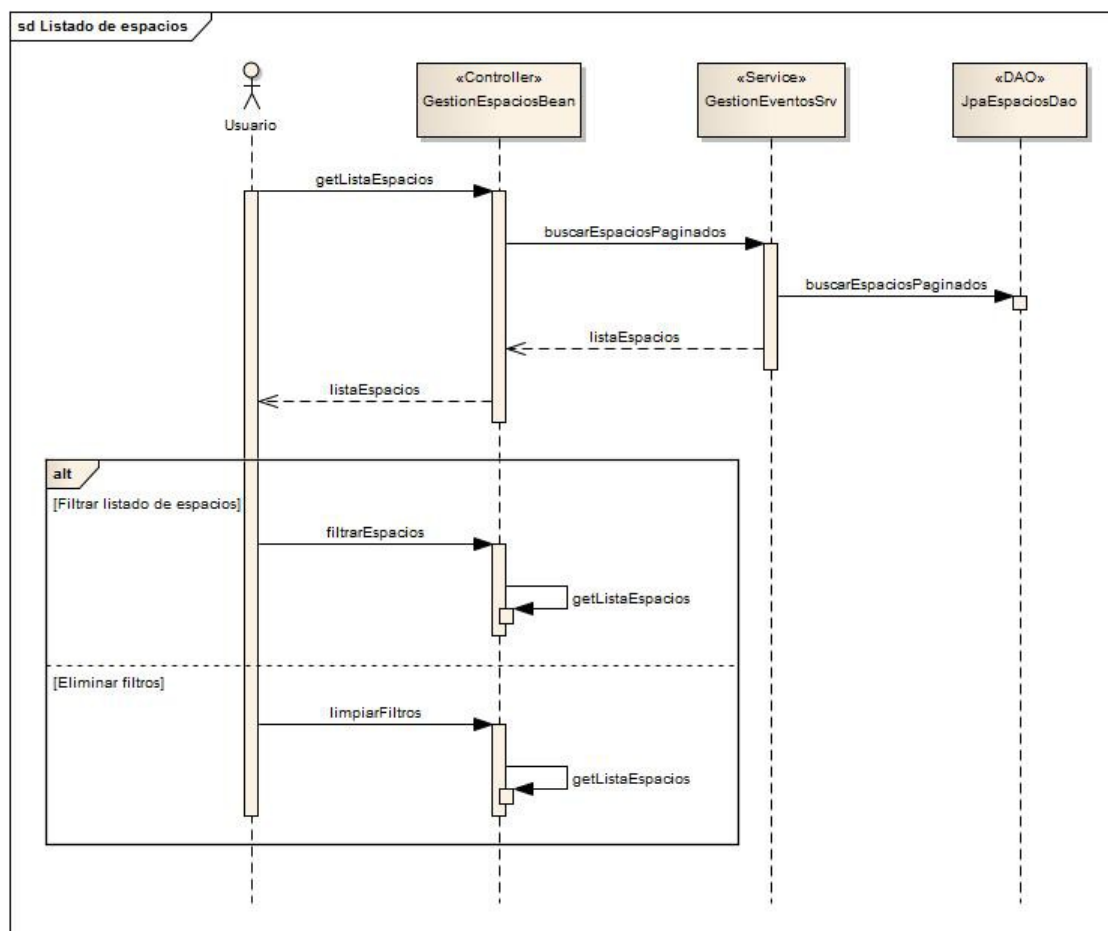
3.4.3.1 Diagrama de secuencia de la creación de espacio



Para la creación de un nuevo espacio, se habrá de realizar la siguiente secuencia:

1. El usuario invoca el método `insertarEspacio` del controlador `EdicionEspacioBean`, desde la página web de gestión de edición de espacio.
2. Se introducen los datos básicos del espacio y, si se desea, se generarán dinámicamente tantos datos descriptivos del espacio como sean necesarios.
3. Se realizan las validaciones necesarias sobre los datos introducidos por el usuario. Si no hay error se continúa y si hay error se muestran los mensajes correspondientes.
4. Se invoca al método de negocio `crearNuevoEspacio` del servicio `GestionEventosSrv`.
5. Se valida que no exista un espacio con el mismo nombre para el grupo seleccionado mediante el método `buscarPorNombreYGrupo` del DAO `JpaEspaciosDao`.
6. Si no existe ningún espacio con el mismo nombre en el grupo se componen los datos dinámicos del espacio y se graba el mismo con el método `grabar` del DAO `JpaEspaciosDao`.
7. Se devuelve el resultado de la operación.
8. Se trata el resultado de la operación y se muestra al usuario el mensaje de resultado de la misma.

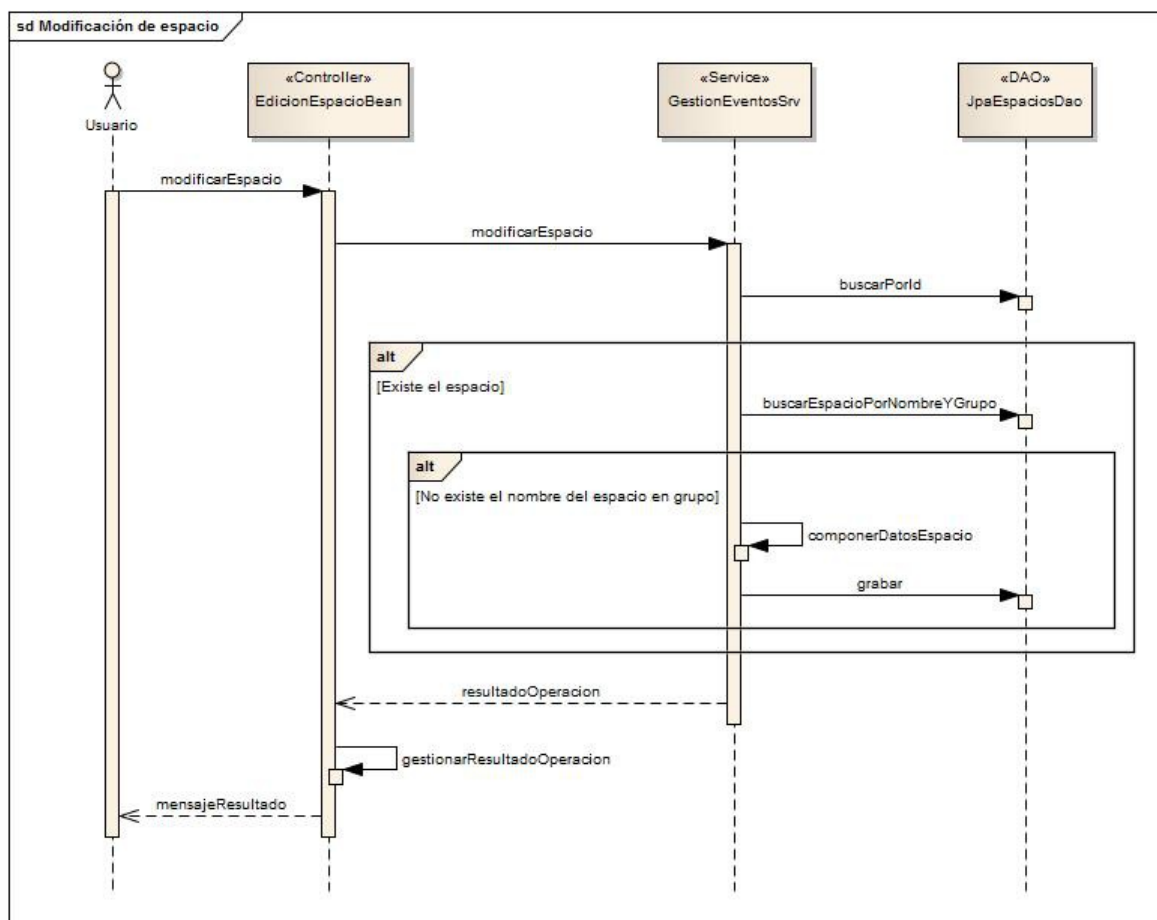
3.4.3.2 Diagrama de secuencia del listado de espacios



Para la listar los contactos de espacios, se realizará la siguiente secuencia:

1. El usuario invoca el método `getListaEspacios` del controlador `GestionEspaciosBean` desde la página web de listado de espacios.
2. Se consulta la primera página de espacios invocando el método `buscarEspaciosPaginados` del servicio `GestionEventosSrv`.
3. Se consulta la página de base de datos mediante el método `buscarEspaciosPaginados` del DAO `JpaEspaciosDao`.
4. Se le muestra al usuario la página del listado o el mensaje de no existencia de espacios. Si hubiese más de una página se puede navegar entre ellas repitiendo la secuencia.
5. Alternativamente se puede filtrar el listado de espacios, con lo que se repetirá la secuencia pero realizando la búsqueda con los filtros introducidos.
6. También se podrán eliminar los filtros, con lo que se repetirá la secuencia tras limpiar los filtros.

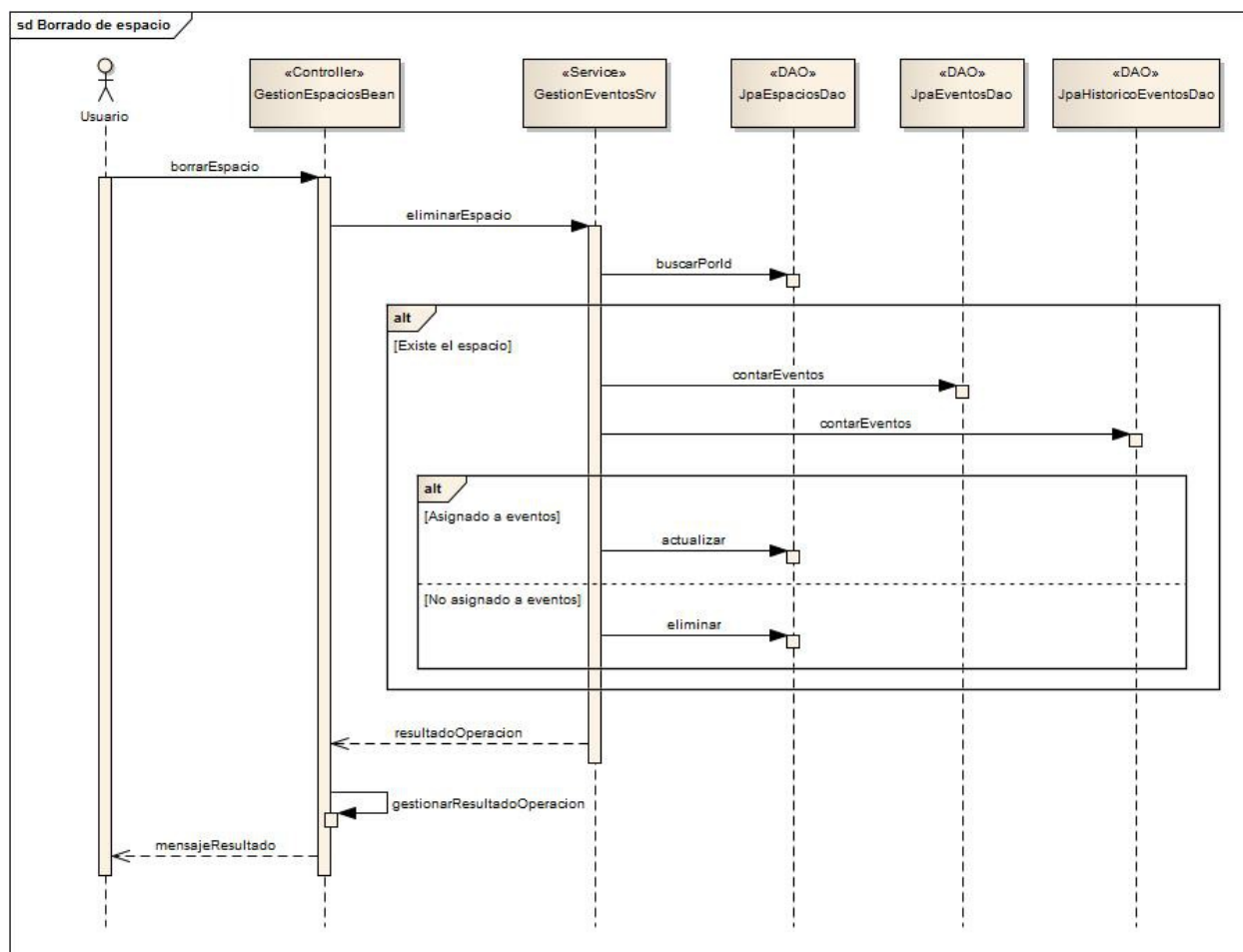
3.4.3.3 Diagrama de secuencia de la modificación de espacio



Para la modificación de un espacio, se realizará la siguiente secuencia:

1. El usuario invoca el método `modificarEspacio` del controlador `EdicionEspacioBean`, desde la página web de edición de espacio tras seleccionar el espacio a modificar del listado de espacios.
2. Se invoca al método `modificarEspacio` del servicio `GestionEventosSrv`.
3. Se comprueba que exista el evento a modificar buscándolo por identificador con el método `buscarPorId` del DAO `JpaEspaciosDao`.
4. Si existe, se comprueba si se ha modificado el nombre del mismo, en cuyo caso habrá que comprobar que no exista un espacio con el mismo nombre en el grupo. Si no se ha modificado el nombre o no existe el nuevo nombre en el grupo, se actualiza el espacio.
5. Se devuelve el resultado de la operación.
6. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje correspondiente.

3.4.3.4 Diagrama de secuencia del borrado de espacio

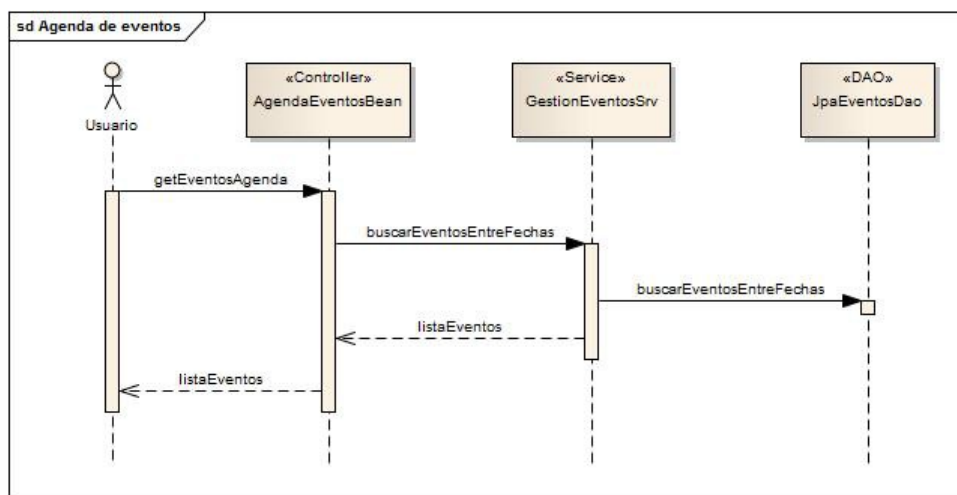


Para el borrado de un espacio, se realizará la siguiente secuencia:

1. El usuario invoca el método `borrarEspacio` del controlador `GestionEspaciosBean`, desde la página web del listado de espacios, seleccionando el espacio del grupo a eliminar.
2. Se invoca al método `eliminarEspacio` del servicio `GestionEventosSrv`.
3. Se comprueba que exista el espacio a eliminar buscándolo por identificador con el método `buscarPorId` del DAO `JpaEspaciosDao`.
4. Si existe se comprueba si tiene eventos o eventos del histórico asignados utilizando sus respectivos DAO, `JpaEventosDao` y `JpaHistoricoEventosDao`.
5. Si tiene eventos asignados y están ya ejecutados se podrá deshabilitar el espacio. Si tiene eventos asignados no ejecutados no se podrá eliminar ni deshabilitar el espacio. Si no tiene eventos asignados se eliminará el espacio.
6. Se devuelve el resultado de la operación.

7. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje correspondiente.

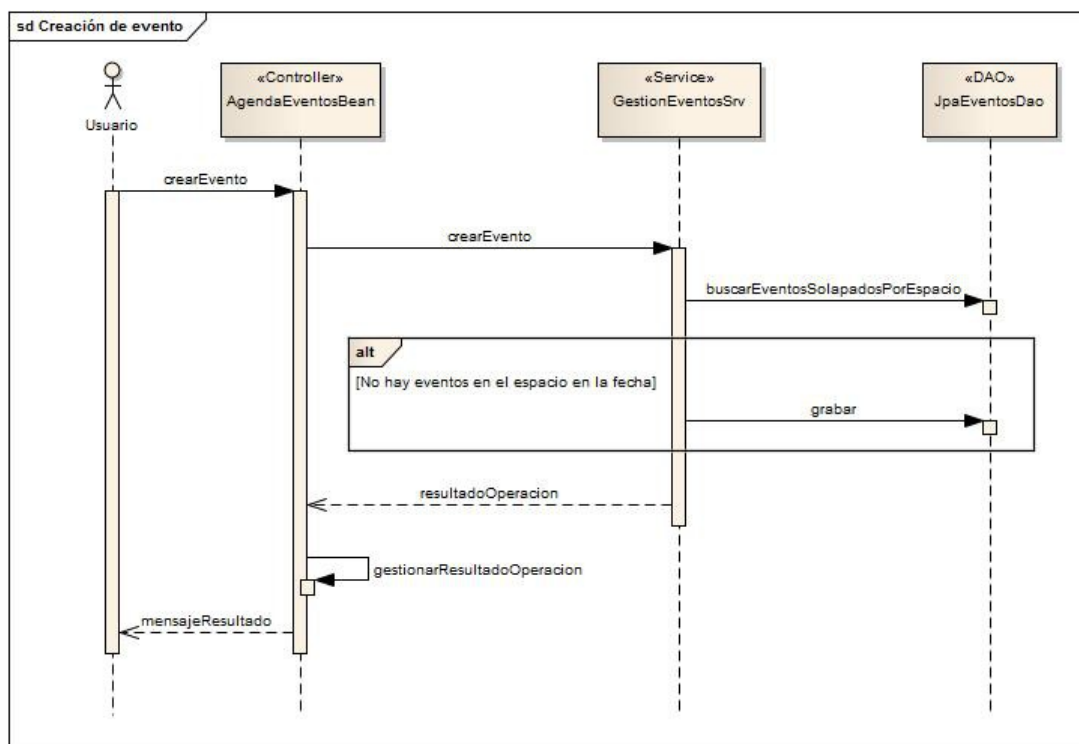
3.4.3.5 Diagrama de secuencia de la agenda de eventos



Para mostrar la agenda de eventos, se habrá de realizar la siguiente secuencia:

1. El usuario invoca el método `getEventosAgenda` del controlador `AgendaEventosBean` desde la página web de la agenda de eventos.
2. Se consultan los eventos del grupo englobados entre las fechas mostradas en la agenda invocando el método `buscarEventosEntreFechas` del servicio `GestionEventosSrv`.
3. Se consultan los eventos mediante el método `buscarEventosEntreFechas` del DAO `JpaEventosDao`.
4. Se le muestra al usuario la agenda de eventos con los eventos encontrados entre las fechas a mostrar.
5. Se podrá navegar por las fechas de la agenda de eventos, con lo que se repetirá la secuencia pero realizando la búsqueda con las nuevas fechas seleccionadas.

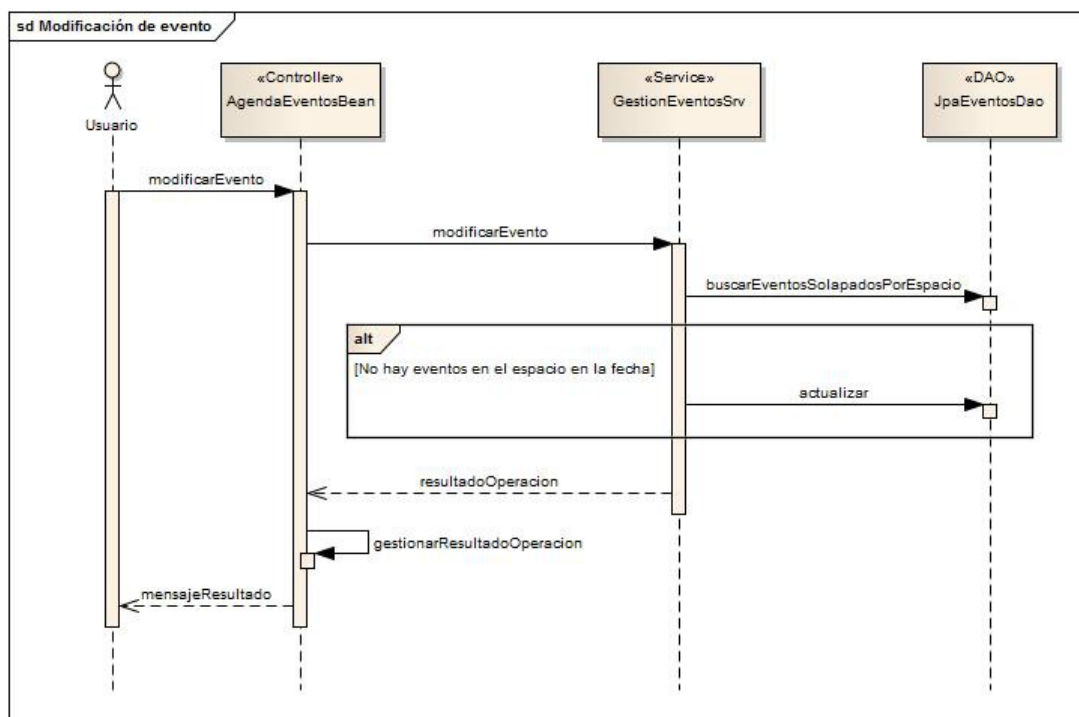
3.4.3.6 Diagrama de secuencia de la creación de evento



Para la creación de un nuevo evento, se habrá de realizar la siguiente secuencia:

1. El usuario invoca el método `crearEvento` del controlador `AgendaEventosBean`, desde la página web de la agenda de eventos.
2. Se introducen los datos básicos del evento, seleccionando el espacio al cual asignar el evento.
3. Se realizan las validaciones necesarias sobre los datos introducidos por el usuario. Si no hay error se continúa y si hay error se muestran los mensajes correspondientes.
4. Se invoca al método de negocio `crearEvento` del servicio `GestionEventosSrv`.
5. Se comprueba que no exista un evento asignado al espacio seleccionado en la fecha que se quiere crear el nuevo evento mediante el método `buscarEventosSolapadosPorEspacio` del DAO `JpaEventosDao`.
6. Si no existe ningún evento asignado al espacio en la fecha seleccionada se inicializa el evento en estado 'Pendiente' y se graba el mismo con el método `grabar` del DAO `JpaEventosDao`.
7. Se devuelve el resultado de la operación.
8. Se trata el resultado de la operación y se muestra al usuario el mensaje de resultado de la misma.

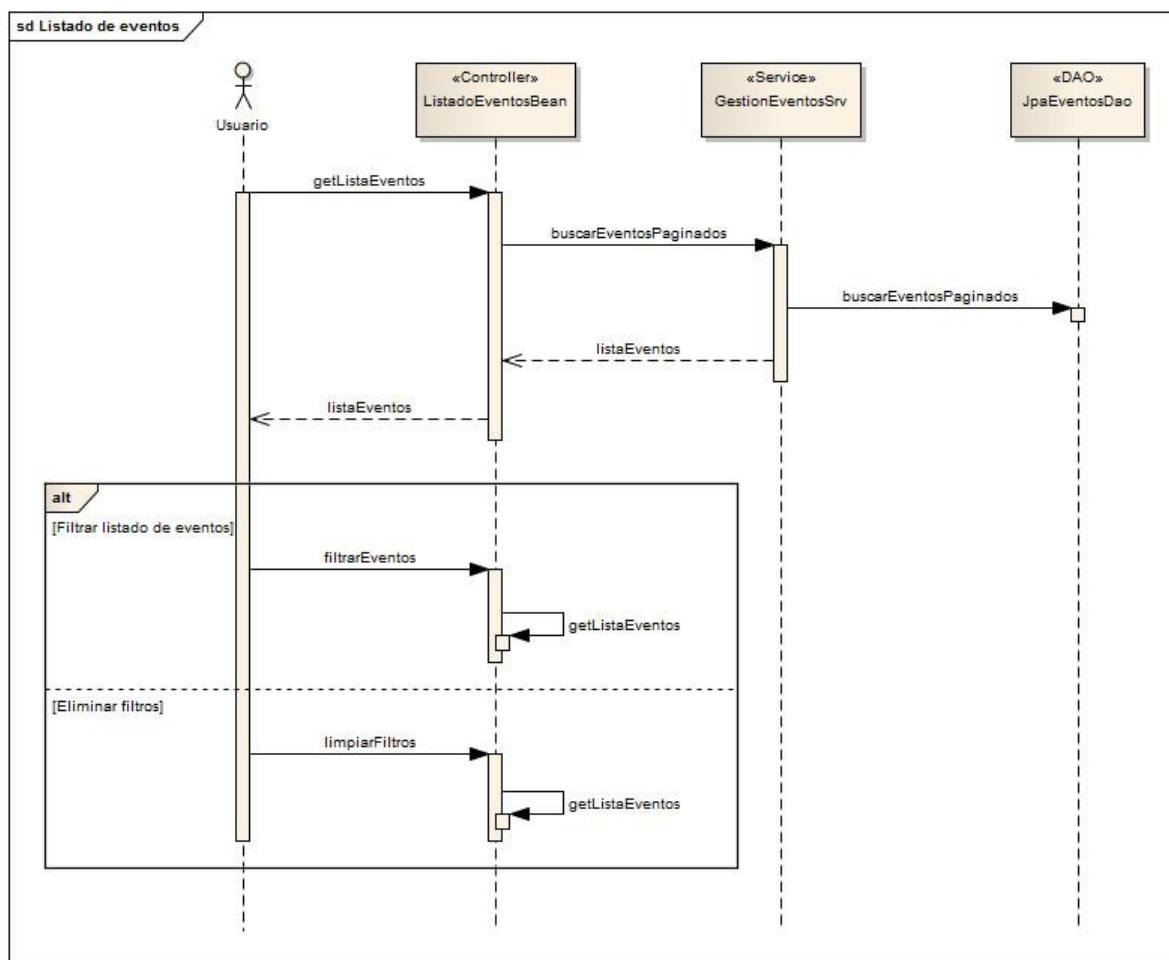
3.4.3.7 Diagrama de secuencia de la modificación de evento



Para la modificación de un evento, se realizará la siguiente secuencia:

1. El usuario selecciona el evento a modificar de la agenda de eventos e invoca el método `modificarEvento` del controlador `AgendaEventosBean`.
2. Se introducen los nuevos datos del evento.
3. Se realizan las validaciones necesarias sobre los datos introducidos por el usuario. Si no hay error se continúa y si hay error se muestran los mensajes correspondientes.
4. Se invoca al método de negocio `modificarEvento` del servicio `GestionEventosSrv`.
5. Se comprueba que no exista un evento asignado al espacio seleccionado en la nueva fecha que se quiere programar el evento mediante el método `buscarEventosSolapadosPorEspacio` del DAO `JpaEventosDao`.
6. Si no existe ningún evento asignado al espacio en la nueva fecha seleccionada se cambia el evento en estado 'Pendiente' y se graba el mismo con el método `grabar` del DAO `JpaEventosDao`.
7. Se devuelve el resultado de la operación.
8. Se trata el resultado de la operación y se muestra al usuario el mensaje de resultado de la misma.

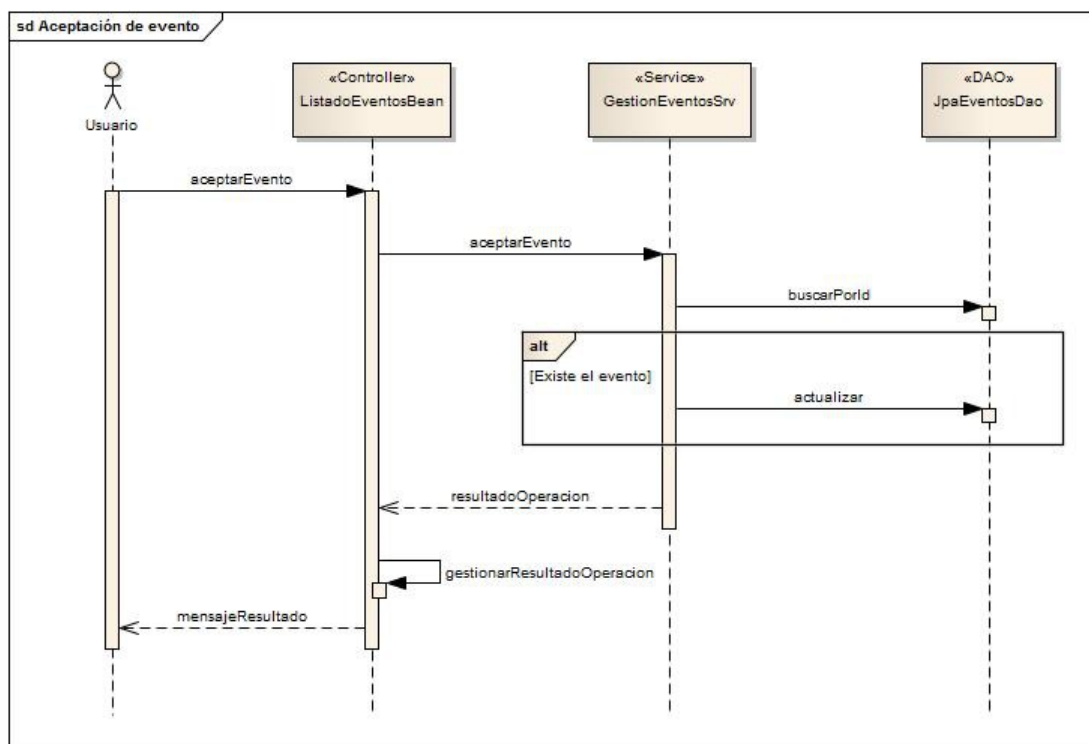
3.4.3.8 Diagrama de secuencia del listado de eventos



Para la listar los eventos programados en un grupo, se realizará la siguiente secuencia:

1. El usuario invoca el método `getListaEventos` del controlador `ListadoEventosBean` desde la página web de listado de eventos.
2. Se consulta la primera página de eventos invocando el método `buscarEventosPaginados` del servicio `GestionEventosSrv`.
3. Se consulta la página de base de datos mediante el método `buscarEventosPaginados` del DAO `JpaEventosDao`.
4. Se le muestra al usuario la página del listado o el mensaje de no existencia de eventos. Si hubiese más de una página se puede navegar entre ellas repitiendo la secuencia.
5. Alternativamente se puede filtrar el listado de eventos, con lo que se repetirá la secuencia pero realizando la búsqueda con los filtros introducidos.
6. También se podrán eliminar los filtros, con lo que se repetirá la secuencia tras limpiar los filtros.

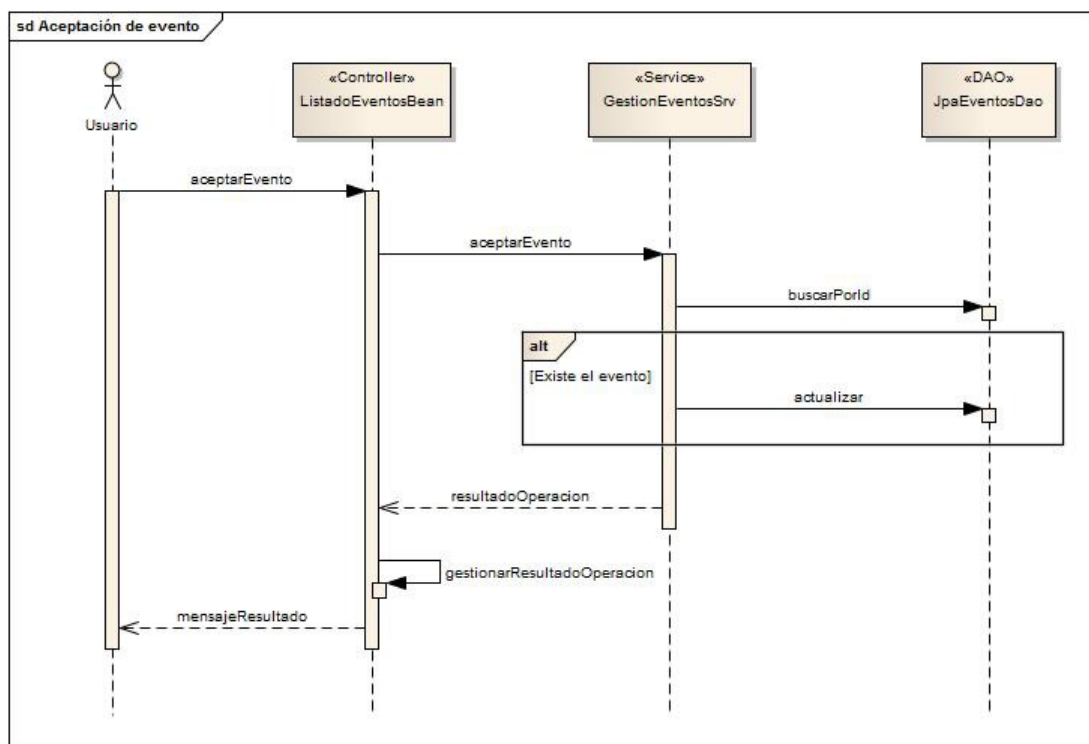
3.4.3.9 Diagrama de secuencia de aceptación de evento



Para aceptar un evento en estado Pendiente o Rechazado, se realizará la siguiente secuencia:

1. El usuario selecciona el evento que se encuentre en estado Pendiente o Rechazado del listado de eventos y cuya fecha de ejecución sea mayor a 15 días de la fecha actual y lo acepta.
2. Se lanza el método `aceptarEvento` del controlador `ListadoEventosBean`.
3. Se invoca al método `aceptarEvento` del servicio `GestionEventosSrv`.
4. Se comprueba que exista el evento a aceptar buscándolo por identificador con el método `buscarPorId` del DAO `JpaEventosDao`.
5. Si existe el evento se pasa al estado Aceptado.
6. Se devuelve el resultado de la operación.
7. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje correspondiente.

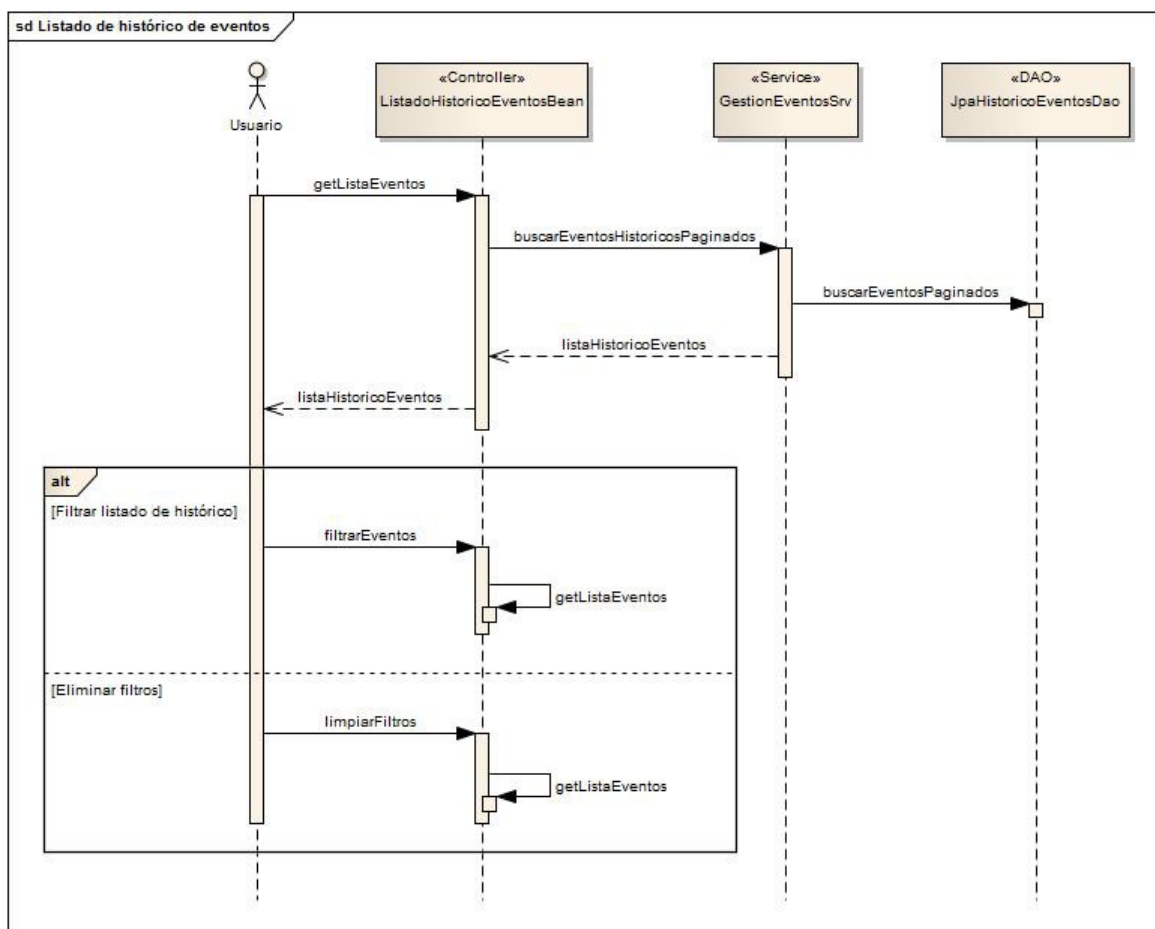
3.4.3.10 Diagrama de secuencia de rechazo de evento



Para rechazar un evento en estado Pendiente, se realizará la siguiente secuencia:

1. El usuario selecciona el evento que se encuentre en estado Pendiente del listado de eventos y lo rechaza.
2. Se lanza el método rechazarEvento del controlador ListadoEventosBean.
3. Se invoca al método rechazarEvento del servicio GestionEventosSrv.
4. Se comprueba que exista el evento a aceptar buscándolo por identificador con el método buscarPorId del DAO JpaEventosDao.
5. Si existe el evento se pasa al estado Rechazado.
6. Se devuelve el resultado de la operación.
7. Se trata el resultado de la operación en el controlador y se devuelve al usuario el mensaje correspondiente.

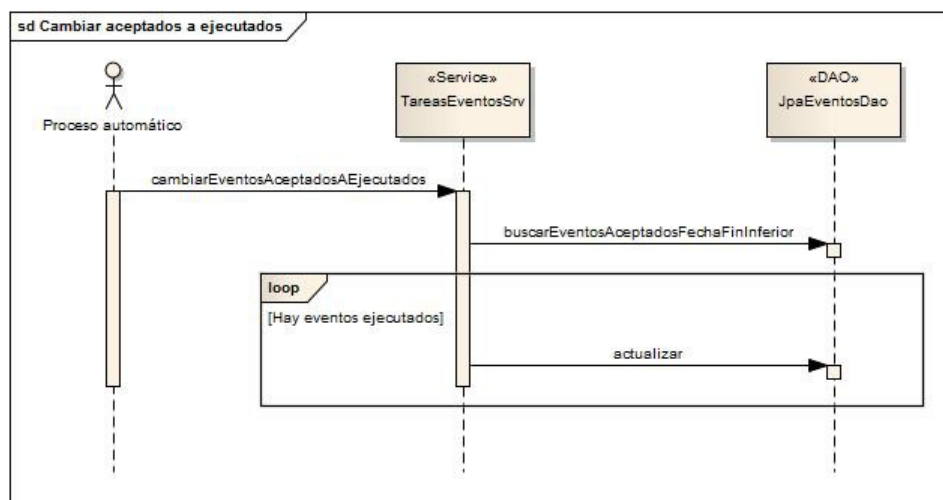
3.4.3.11 Diagrama de secuencia del listado del histórico de eventos



Para la listar el histórico eventos ejecutados en un grupo, se realizará la siguiente secuencia:

1. El usuario invoca el método `getListaEventos` del controlador `ListadoHistoricoEventosBean` desde la página web de listado del histórico de eventos.
2. Se consulta la primera página de eventos invocando el método `buscarEventosHistoricosPaginados` del servicio `GestionEventosSrv`.
3. Se consulta la página de base de datos mediante el método `buscarEventosPaginados` del DAO `JpaHistoricoEventosDao`.
4. Se le muestra al usuario la página del listado o el mensaje de no existencia de eventos. Si hubiese más de una página se puede navegar entre ellas repitiendo la secuencia.
5. Alternativamente se puede filtrar el listado del histórico de eventos, con lo que se repetirá la secuencia pero realizando la búsqueda con los filtros introducidos.
6. También se podrán eliminar los filtros, con lo que se repetirá la secuencia tras limpiar los filtros.

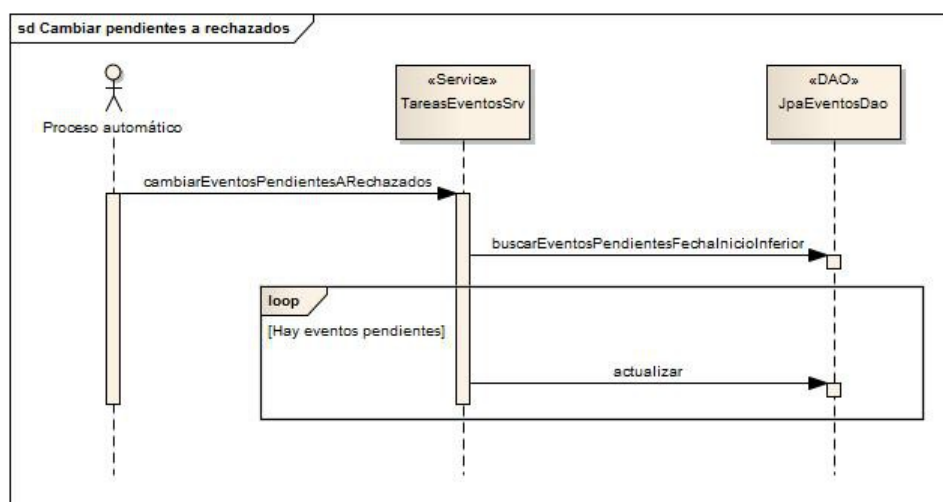
3.4.3.12 Diagrama de secuencia del paso de eventos a ejecutados



Para la pasar automáticamente los eventos aceptados ya ejecutados al estado Ejecutado, se realizará la siguiente secuencia:

1. El proceso automático comprueba que sea la fecha de lanzamiento de la tarea y si es así se ejecuta.
2. Se invoca el método `cambiarEventosAceptadosAEjecutados` del servicio `TareasEventosSrv`.
3. Se buscan los eventos aceptados cuya fecha de fin sea anterior a la fecha actual con el método `buscarEventosAceptadosFechaFinInferior` del DAO `JpaEventosDao`.
4. Se recorren los eventos encontrados y se actualiza su estado a Ejecutado.

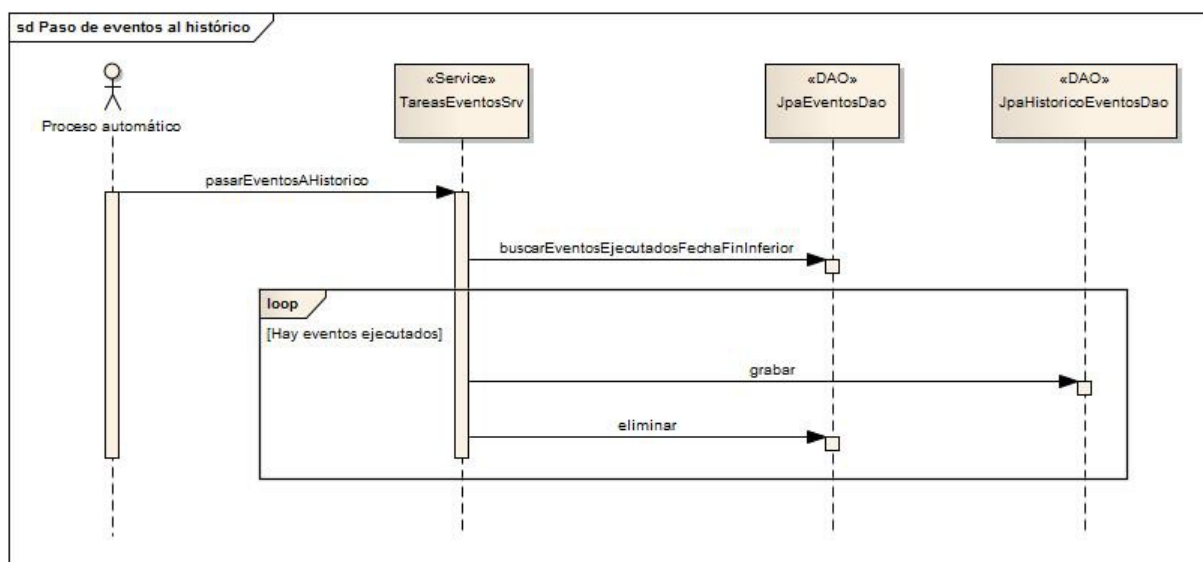
3.4.3.13 Diagrama de secuencia del paso de eventos a rechazados



Para la pasar automáticamente los eventos pendientes al estado Rechazado, se realizará la siguiente secuencia:

1. El proceso automático comprueba que sea la fecha de lanzamiento de la tarea y si es así se ejecuta.
2. Se invoca el método `cambiarEventosPendientesAREchazados` del servicio `TareasEventosSrv`.
3. Se buscan los eventos pendientes cuya fecha de ejecución sea anterior a 15 días desde la fecha actual con el método `buscarEventosPendientesFechaInicioInferior` del DAO `JpaEventosDao`.
4. Se recorren los eventos encontrados y se actualiza su estado a Rechazado.

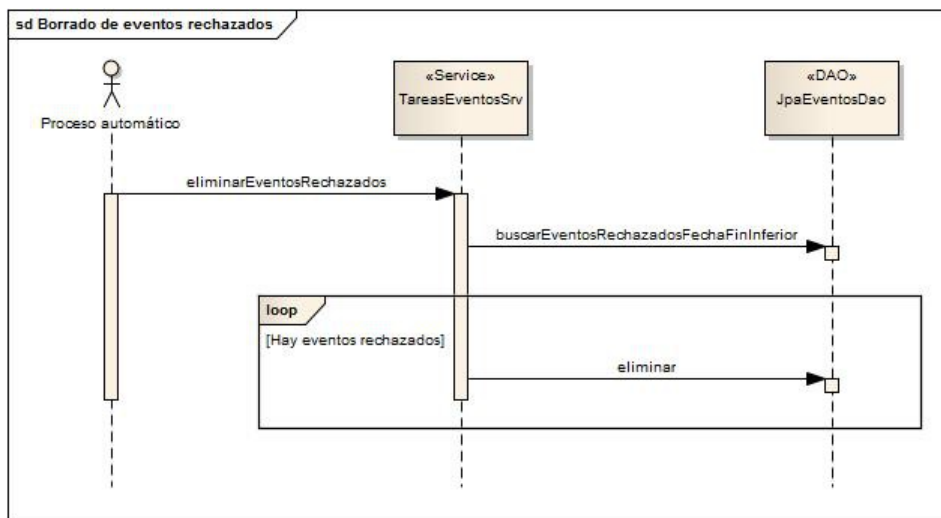
3.4.3.14 Diagrama de secuencia del paso de eventos al histórico



Para la pasar automáticamente los eventos ejecutados al histórico de eventos, se realizará la siguiente secuencia:

1. El proceso automático comprueba que sea la fecha de lanzamiento de la tarea y si es así se ejecuta.
2. Se invoca el método `pasarEventosAHistorico` del servicio `TareasEventosSrv`.
3. Se buscan los eventos ejecutados cuya fecha de fin sea anterior a un año desde la fecha actual con el método `buscarEventosEjecutadosFechaFinInferior` del DAO `JpaEventosDao`.
4. Se recorren los eventos encontrados, se graban en la tabla de histórico de eventos ejecutados y se eliminan de la tabla de eventos.

3.4.3.15 Diagrama de secuencia del borrado de eventos rechazados



Para la eliminar automáticamente los eventos rechazados antiguos, se realizará la siguiente secuencia:

1. El proceso automático comprueba que sea la fecha de lanzamiento de la tarea y si es así se ejecuta.
2. Se invoca el método `eliminarEventosRechazados` del servicio `TareasEventosSrv`.
3. Se buscan los eventos rechazados cuya fecha de fin sea anterior a la fecha actual con el método `buscarEventosRechazadosFechaFinInferior` del DAO `JpaEventosDao`.
4. Se recorren los eventos encontrados y se eliminan de la tabla de eventos.



4 Tecnologías utilizadas

4.1 – Frameworks utilizados

4.2 – Entorno de desarrollo

El objetivo de este apartado será analizar la arquitectura diseñada para el sistema GEEC, si es la más adecuada para la explotación del sistema por el usuario final o las personas designadas por el mismo para tal fin y si se trata de la arquitectura adecuada para construir el sistema de gestión de espacios y eventos culturales, ya que uno de los requisitos a la hora de implementar el mismo es la creación de una arquitectura de desarrollo. La meta, a la hora de crear la arquitectura, es diseñarla en su totalidad utilizando Software Libre, lo cual facilitará la evolución del sistema debido a los beneficios que este puede aportar al trabajar con código abierto.

En el caso de la aplicación GEEC, al decidir realizar el desarrollo utilizando el lenguaje de programación Java, se utilizarán las tecnologías más extendidas a la hora de realizar desarrollos web, intentando conseguir que el código resultante sea sencillo y se encuentre bien estructurado, lo cual permitirá la continuación o evolución del desarrollo.

4.1 Frameworks utilizados

En este apartado se analizará la arquitectura utilizada para la implementación del sistema, así como los frameworks y/o tecnologías utilizadas para la creación de la misma.

Como se ha visto en apartados anteriores, se aplicarán distintos patrones de diseño a la hora de diseñar la arquitectura, lo cual permitirá dividir la implementación de la aplicación en distintas capas en relación con la tarea a desarrollar en cada una de ellas. Esto permitirá desacoplar cada una de ellas y a su vez los distintos frameworks utilizados para diseñar la arquitectura, con lo que la migración o modificación del framework utilizado a una capa no será una tarea imposible de realizar en futuras evoluciones del sistema. Además, facilitará el mantenimiento del mismo, al englobar la lógica de negocio en determinadas áreas de la arquitectura.

En concreto, para la creación de la arquitectura se utilizarán los patrones de diseño **Modelo-Vista-Controlador**, **Service** y **DAO**, utilizando distintas tecnologías para la implementación de cada uno de ellos. La utilización de estos patrones de diseño, permitirán dividir la aplicación en varias capas lógicas.

La aplicación del patrón **Modelo-Vista-Controlador**, permitirá diseñar la capa de presentación y el modelo de objetos en tres partes: por un lado se tendrá el modelo de objetos diseñado durante la fase de análisis que se utilizará para pasar la información a través de las distintas capas de la arquitectura y que



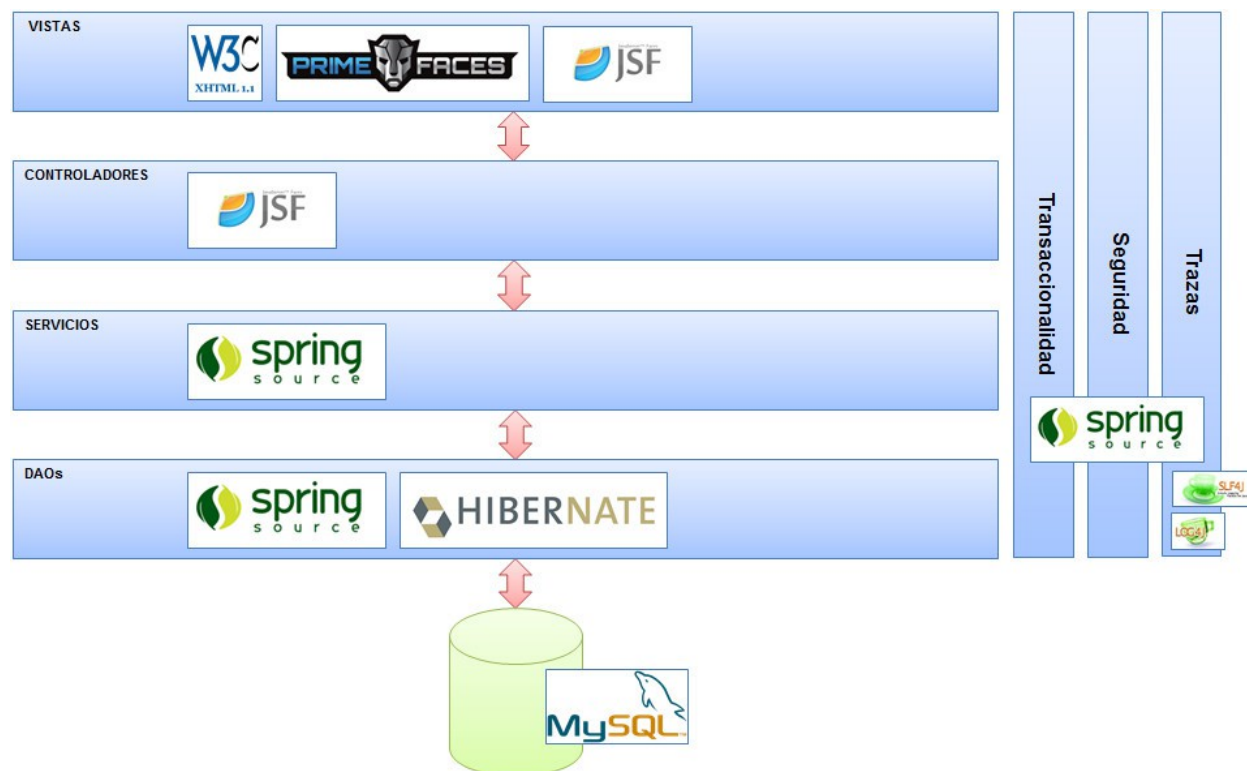
se implementará con POJOs Java y no contendrán ningún código relativo a la lógica a implementar. Para las capas de vista y controlador se utilizará el framework JSF (Java Server Faces), utilizando las tecnologías JSP y XHTML para la implementación de las vistas o presentaciones, junto con las librerías de componentes proporcionadas por JSF y el framework Primefaces, y clases Java para implementar los controladores que apoyarán a las capas de presentación.

La utilización del patrón **DAO** (Data Access Object) permitirá encapsular y abstraer todo el acceso a las fuentes de datos utilizadas por el sistema. Esta capa de la arquitectura se encargará de administrar la conexión con el origen de datos para obtener y almacenar los mismos. Si el sistema necesitase utilizar distintas fuentes de datos, mediante el patrón **DAO** se implementará la lógica de acceso para cada una de ellas sin afectar al código de los componentes que representan la lógica de negocio, es decir, los servicios. Los componentes de negocio accederán a los objetos DAO utilizando interfaces sencillas que encapsulen los mecanismos de obtención y recuperación de datos. Debido a este principio (denominado **programación orientado a interfaces**), si las fuentes de datos cambiasen, tanto de proveedor como de tipo, los servicios seguirían utilizando las mismas interfaces y no será necesario cambiarlos, ya que lo único que sería necesario es implementar nuevos objetos DAO relativos al nuevo tipo de fuente de datos. Esta capa se implementará siguiendo la especificación JPA (Java Persistence API) y la implementación concreta del framework Hibernate.

La aplicación del patrón **Services**, aplica el paradigma de diseño orientado a servicios, el cual permitirá organizar un inventario de servicios de lógica de negocio dentro de la capa de servicios. Al agrupar la lógica de negocio en los servicios, se reducirá el impacto del cambio sobre la aplicación, ya que los cambios sólo afectarán a la capa en los que se encuentren dichos servicios sin afectar al resto de capas, simplificando además el mantenimiento del sistema. Esta capa se implementará utilizando el framework Spring. El principio de programación **orientado a interfaces** también se aplicará en la capa de servicios.

Hay que tener en cuenta que, algunos elementos comunes de las aplicaciones como la transaccionalidad, la seguridad y las trazas no se han abarcado con los patrones y frameworks anteriores. Esto se debe a que se tratan de funcionalidades transversales a toda la aplicación, por lo que se utilizará **AOP** (Aspect Oriented Programming), programación orientada a aspectos, un paradigma de programación que permitirá dividir la lógica de programación en distintas áreas concretas por tipo de funcionalidad que proporcionan, y que afectan a distintas áreas de la aplicación. Para la implementación de AOP se utilizará también el framework Spring, el cual permitirá aplicar AOP a la gestión transaccional y de trazas y Spring Security, el cual permitirá aplicar AOP en el área de seguridad, y en concreto en la autenticación y autorización, del sistema.

La siguiente imagen describirá la arquitectura lógica diseñada y cómo y qué frameworks se aplicarán en cada una de las mismas:



4.1.1 JSF (JavaServer Faces) y componentes Primefaces

JavaServer Faces es un framework diseñado para simplificar la construcción de interfaces de usuario de aplicaciones web, permitiendo crear componentes reutilizables de interfaz de usuario y conectando dichos componentes con el código de servidor de la aplicación y enlazar los eventos generados en la parte cliente, el navegador web, con los manejadores de eventos que se encontrarán en la parte servidora.

JSF proporciona además dos librerías de etiquetas personalizadas para páginas JSP (JavaServer Pages), aunque permite utilizar otras tecnologías para la representación de la vista, que incluyen los elementos estándar de HTML.

Además de los componentes estándar de JSF, se incluirá la librería PrimeFaces, la cual proporciona un conjunto extendido de componentes que integran JQuery y otras tecnologías muy útiles a la hora de implementar la interfaz de usuario. Al no requerir configuración adicional para su utilización permite una fácil integración con otras librerías de componentes JSF como IceFaces o RichFaces.

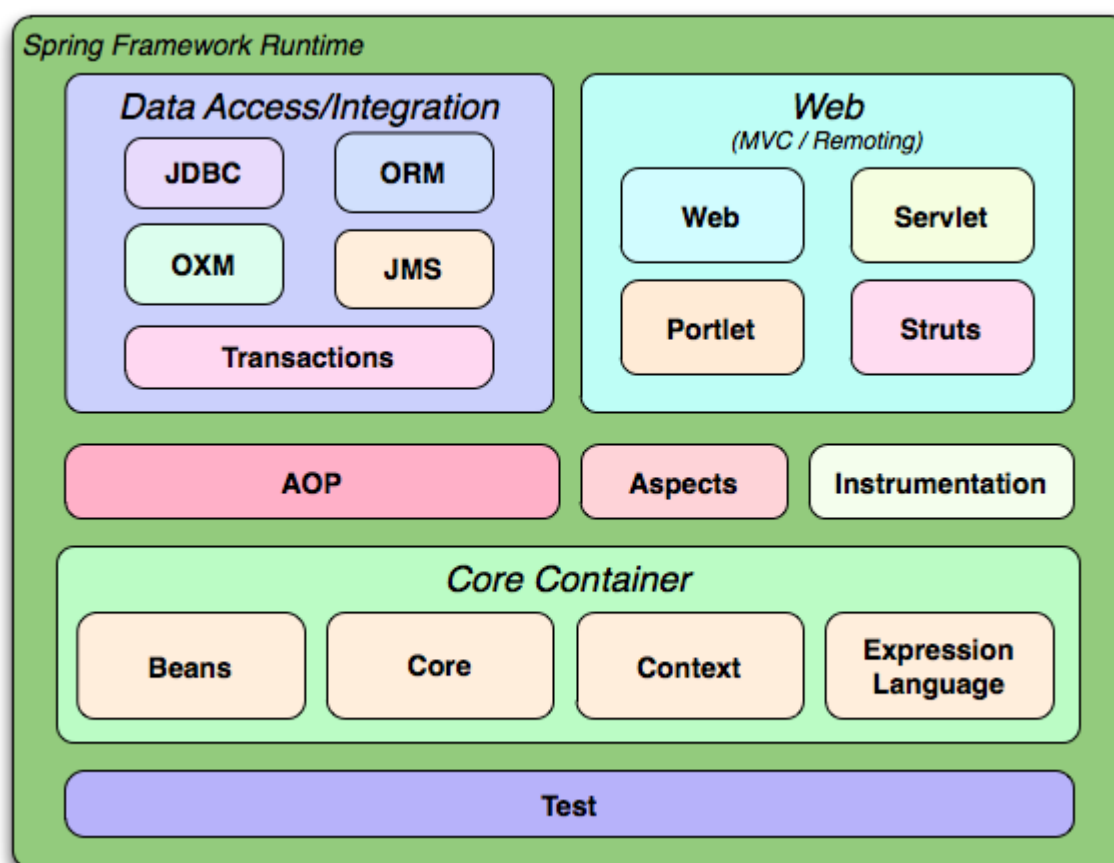
4.1.2 Spring

El framework Spring es una plataforma que proporciona la infraestructura que actúa de soporte para



desarrollar aplicaciones Java. La principal característica de Spring es que permite controlar la infraestructura del sistema a nivel de la aplicación, es decir, encapsula toda la funcionalidad estándar de las aplicaciones JEE para que los equipos de desarrollo puedan centrarse en la lógica de negocio de la aplicación. Spring proporciona, entre otras cosas, inyección de dependencias, programación orientada a aspectos, gestión transaccional, cacheo, validación, abstracción para trabajar con especificaciones JEE como JDBC y JPA, funcionalidades para la realización de pruebas unitarias y de integración, etc...

Debido a su diseño modular permite utilizar independientemente partes individuales del framework según sea necesario. Aunque dispone de aproximadamente 20 módulos, sin contar proyectos independientes como Spring Security, en la implementación del sistema GEEC no se van a utilizar todos ellos, por lo que sólo se describirán los utilizados.



- **Core:** El módulo Core de Spring proporciona la funcionalidad básica del framework. Se utiliza para la configuración de la aplicación mediante el uso de la inyección de dependencias, lo cual permite desacoplar la configuración de las dependencias entre clases sin que afecte a la lógica de negocio de la aplicación.
- **Context:** El módulo Context es el encargado de extender ciertas funcionalidades proporcionadas por el Core y de aportar diversos servicios como gestión de correo electrónico, acceso JNDI



(Java Naming and Directory Interface), integración con EJB (Enterprise Java Beans), remoting o scheduling (para programación de tareas).

- **AOP:** El módulo AOP proporciona el API de programación orientada a aspectos proporcionar funcionalidades transversales a las distintas capas lógicas de la aplicación. Este módulo es utilizado por otros como **Transaction**, etc...
- **Transactions:** El módulo Transactions proporciona una capa de abstracción para la gestión de transacciones con gestión declarativa de transacciones, un API de gestión programática de transacciones y un modelo consistente de programación entre las distintas APIs transaccionales como JTA, JDBC, etc...
- **ORM:** El módulo ORM (Object Relational Mapping) proporciona los mecanismos de integración con frameworks de persistencia mediante mapeo objeto relacional como JPA, Hibernate y myBatis entre otros.
- **Web:** El módulo web proporciona las funcionalidades de integración orientadas a proyectos web, tales como carga de ficheros en servidor, mapeo de parámetros del formulario a objetos, etc... Además proporciona utilidades de integración con distintos frameworks de presentación como JSF, Struts, etc...
- **Expression:** El módulo Expression proporciona el soporte del lenguaje de expresiones de Spring que permite manipular y consultar un grafo de objetos en tiempo de ejecución.

4.1.3 Spring Security

Spring Security, aunque englobado dentro del framework Spring, se ha convertido en un proyecto independiente. Proporciona seguridad declarativa para aplicaciones basadas en el framework Spring, ofreciendo una solución integral de seguridad, de control de autenticación y autorización tanto a nivel de peticiones HTTP como a nivel de métodos de clases.

Este framework es un gestor de seguridad en el que destaca su versatilidad proporcionando un amplio conjunto de servicios de autenticación y control de acceso para las aplicaciones en las que se utiliza, recubriendo diversos estándares de seguridad de Java. Spring Security proporciona la funcionalidad necesaria para adoptar mecanismos de seguridad a aplicaciones Java utilizando características de programación orientada a aspectos de forma transparente para el desarrollador mediante el soporte del framework Spring.

Entre las características y ventajas más importantes de Spring se hallan la instalación no intrusiva mediante filtros web, mantiene los objetos libres de código del framework, posee una arquitectura flexible, soporta certificados X.509, soporta la integración con LDAP, cifrado de datos, integración con las



bases de datos existentes, etc...

4.1.4 JPA e Hibernate

JPA o Java Persistence API es un API de persistencia Java para gestionar datos relacionales utilizando el lenguaje de programación Java. La especificación JPA se divide en tres áreas concretas:

- El propio API de persistencia
- El JPQL (Java Persistence Query Language), un lenguaje de consulta diseñado como extensión orientada a objetos de SQL para proporcionar la conexión entre el mundo de objetos y el mundo relacional e independencia del lenguaje SQL específico de cada sistema gestor de base de datos.
- El API criteria para consulta
- Los metadatos objeto/relacionales

El objetivo perseguido por el API es no perder las ventajas de la programación orientada a objetos al interactuar con bases de datos y permitir la utilización de los objetos básicos conocidos en Java como POJOs.

Aunque existen muchas implementaciones del API JPA como Hibernate, EclipseLink, OpenJPA, etc... pero para la implementación del sistema de gestión de eventos y espacios utilizaremos Hibernate, siempre siguiendo la especificación del API para no acoplar el desarrollo y futuros desarrollos a la utilización de este framework.

Hibernate es un framework de persistencia que permite mapear un modelo de clases con un modelo relacional sin imponer restricciones en ninguno de los diseños. Permite desarrollar clases persistentes a partir de clases comunes, incluyendo la asociación, la herencia, el polimorfismo, la composición y colecciones de objetos.

Soporta todos los sistemas gestores de bases de datos relacionales y se integra sin restricciones con los contenedores web y servidores de aplicaciones más extendidos. Las principales características de Hibernate son:

- Puede operar proporcionando persistencia de forma transparente para el desarrollador.
- Soporta de una forma natural el paradigma de orientación a objetos, con herencia, polimorfismo, composición y colecciones.
- Permite gran variedad de mapeos para colecciones y objetos dependientes.
- Posee un alto rendimiento, puede proporcionar dos niveles de caché y ser utilizado en cluster.



- Permite carga perezosa (lazy) de objetos.
- Soporta los distintos tipos de generación de identificadores proporcionados por los sistemas gestores de base de datos como secuencias, autoincrementales, etc...

4.1.5 JSON y Jackson

A la hora de crear el sistema de gestión de espacios y eventos culturales, se encuentra la problemática de que los espacios a gestionar pueden ser muy diversos, desde salas de teatro a salas de concierto o simples salas de reunión. Como se desea almacenar información estructural del espacio y estos son muy diversos, además de que el almacenamiento de esta información para los distintos espacios es opcional, es imposible diseñar el modelo relacionar de forma que cubra todas las posibilidades que se pueden dar sin perder rendimiento en el sistema gestor de base de datos y haciendo que el diseño del mismo deje de ser óptimo.

Por esta razón, la tabla de base de datos sólo contendrá un campo denominado detalles técnicos, donde se almacenará la descripción técnica del espacio a gestionar como dimensiones, mobiliario, etc... Como dentro de la variabilidad de los detalles técnicos de los distintos espacios sí podemos encontrar elementos comunes como dimensiones de distintas partes o componentes del mismo, se buscará implementar una opción que permita gestionar de forma elegante esta descripción técnica.

Es aquí donde entra en juego JSON (JavaScript Object Notation), un estándar abierto basado en texto diseñado para el intercambio de datos legibles. Este estándar proviene del lenguaje de scripting utilizado en JavaScript para representar estructuras de datos simples y colecciones, llamados objetos. Independientemente de su relación con JavaScript, es un lenguaje independiente y posee procesadores en múltiples lenguajes, entre ellos Java, el lenguaje elegido para la implementación del sistema. Aunque es similar a XML, el hecho de que JSON se utilice cada vez más en relación con Ajax para solicitar o dibujar datos en una página web una vez esta ha sido cargada en el navegador, esto permitirá modificar de forma dinámica la interfaz de usuario según se añadan dinámicamente nuevos datos sobre el espacio por parte del usuario, sin necesidad de mantener una estructura rígida de almacenamiento en base de datos.

Aunque JSON está basado en texto y por tanto se puede generar utilizando únicamente cadenas de texto, en el sistema GEEC se utilizará la librería Jackson disponible en Java. Jackson es una librería Java multipropósito para procesar JSON rápida, ligera y muy sencilla de utilizar para los desarrolladores debido a la posibilidad de realizar Data Binding, es decir convertir JSON a objetos POJO y viceversa en función de sus métodos de acceso o de anotaciones.

Utilizando este sistema, los datos variables relativos a los espacios gestionados por el sistema se almacenarán en base de datos como cadenas de texto, pero podrán representarse en pantalla al usuario como distintos campos de entrada de datos dinámicos, es decir, podrá introducir tantos o tan pocos



elementos como le sea necesario, permitiendo así abarcar una gran variedad de descripciones técnicas distintas sin llenar la base de datos de campos vacíos.

Un ejemplo de detalles técnicos de un espacio podría ser:

```
{
  "tipoRecinto": "Cubierto",
  "dimensiones": {
    "ancho": 25,
    "alto": 17,
    "fondo": 29
  },
  "aforo": [
    {
      "nombre": "Balcón central",
      "cantidad": 20
    },
    {
      "nombre": "Balcón derecho",
      "cantidad": 10
    },
    {
      "nombre": "Balcón izquierdo",
      "cantidad": 10
    },
    {
      "nombre": "Patio",
      "cantidad": 60
    }
  ],
  "escenario": {
    "disposicion": "Anfiteatro",
    "pendiente": 0
  }
}
```

4.1.6 SLF4J y Log4Java

La librería de Apache, Log4Java es una de las librerías de código abierto de referencia para los aspectos de traza durante el desarrollo y/o explotación de aplicaciones, permitiendo además elegir el nivel de detalle de los mensajes de traza en tiempo de ejecución sin tener que tocar el código compilado, sino únicamente ficheros de configuración.

Aplicando el mismo principio que durante el desarrollo de las capas de Servicios y DAOs, se utilizará SLF4J (Simple Logging Facade for Java) como abstracción para el framework de traza que se utilice, en el caso del sistema GEEC Log4Java, lo cual permitirá en futuras evoluciones de la aplicación que el usuario seleccione el framework de traza en tiempo de despliegue.

4.1.7 Junit y EasyMock

Durante el desarrollo de software es necesario ejecutar pruebas tanto unitarias como de integración para verificar que el trabajo está realizándose como estaba previsto. Además, según aumenta el tamaño del sistema, aumentan las probabilidades de introducir un elemento que ocasione errores en el código ya existente.

Mediante la ejecución de pruebas unitarias se podrán lanzar todas las pruebas necesarias cuando se introduzca un cambio sobre un método o clase, para asegurar que no se ha alterado funcionamiento normal del código ya existente. Durante la implementación del proyecto se implementarán pruebas unitarias a nivel de capa, pudiendo comprobar que se ejecuta correctamente el funcionamiento de los



servicios y los DAO. Una vez ejecutadas correctamente las pruebas unitarias de cada uno de los objetos de cada capa, se ejecutarán las pruebas de integración sobre los objetos de las distintas capas trabajando de forma colaborativa.

Como apoyo de JUnit para la creación de objetos mock que simulen la ejecución de código, por ejemplo de acceso a datos, se utilizará EasyMock.

Ambas librerías se integran perfectamente con Spring para la ejecución de los tests.

4.2 Entorno de desarrollo

Para la organización del entorno de desarrollo del sistema, se seguirá la misma política de uso de software libre, en la medida de lo posible. Se intentará que todas las herramientas utilizadas sean de código abierto o en su defecto, que dispongan de licencias que no obliguen al pago por el uso de las mismas.

El entorno de desarrollo a utilizar precisará de un Entorno de Desarrollo propiamente dicho, un servidor de aplicaciones o contenedor servlet y un sistema gestor de base de datos, para los cuales se utilizará respectivamente Eclipse IDE, Apache Tomcat y MySQL. Además se utilizará Apache Maven como sistema de construcción de proyectos.

4.2.1 Eclipse IDE

Eclipse es un IDE (Integrated Development Environment) o Entorno de Desarrollo Integrado multiplataforma de código abierto basado en Java liderado por la Fundación Eclipse. En sí mismo, Eclipse es un marco y conjunto de servicios creados sobre la plataforma OSGi para construir un entorno de desarrollo a partir de plug-ins que proporcionan distintas funcionalidades como desarrollo Java (JDT, Java Development Tools) proyectos web (WTP, Web Tools Project), etc...

4.2.2 Apache Tomcat

Apache Tomcat es un contenedor servlet de código abierto cuyo proyecto está liderado por la Apache Software Foundation. A pesar de ser un proyecto de software libre, es la implementación de referencia para distintas especificaciones de la Java Community Process y es utilizado como contenedor servlet en entornos con un alto nivel de tráfico en los que además es necesaria alta disponibilidad.

Además Eclipse permite integrarse completamente con Apache Tomcat para poder desplegar y depurar aplicaciones desarrolladas en este entorno de desarrollo.



4.2.3 Sistema gestor de base de datos MySQL

MySQL es un sistema gestor de bases de datos (SGBD) relacionales, multihilo y multiusuario. Es el SGBD de código abierto más popular hasta la fecha. Sus características principales son:

- Es un sistema de gestión de base de datos relacionales que permite añadir, acceder y procesar los datos almacenados en sus bases de datos.
- Es Open Source, lo cual permite que cualquiera utilice y modifique su código.
- Es rápido, fiable y fácil de utilizar, por lo que ha sido utilizado con éxito en entornos de producción de alto rendimiento.
- Trabaja en entornos cliente/servidor o incrustados.

4.2.4 Apache Maven

Apache Maven surge de la necesidad de realizar tareas comunes con cierta frecuencia en distintos proyectos Java. Permite, mediante una serie de comandos sencillos, crear la estructura de directorios, gestionar las dependencias de nuestro proyecto con otras librerías, compilar el proyecto, generar el empaquetado, la documentación y diversos informes de análisis de código. Además también se encarga de pasar automáticamente los test de prueba. En definitiva, se trata de una herramienta software para la gestión y construcción de proyectos Java.

Maven utiliza un fichero de configuración XML denominado Project Object Model (POM), para describir el proyecto a construir, sus dependencias de otros módulos y componentes externos y el orden de construcción de los mismos. Tiene una gran variedad de plugins predefinidos que permiten realizar ciertas tareas claramente definidas como la compilación de código, empaquetado, documentación, generación de métricas, etc...

En el desarrollo del sistema GEEC, Maven permitirá agilizar el tiempo de desarrollo del mismo.



5 Conclusiones y futuros trabajos

6.1 – Objetivos (cumplidos)

6.2 – Futuros trabajos

Una vez finalizado el desarrollo del proyecto, se extraerán una serie de conclusiones sobre la realización del proyecto GEEC, lo que ha supuesto el desarrollo del mismo, los objetivos que se han alcanzado y que han quedado pendientes de alcanzar, y se analizarán con detenimiento los futuros trabajos que se podrían realizar sobre el sistema para extender su funcionalidad.

5.1 Objetivos (cumplidos)

Tal y como se describió en la introducción de esta memoria, el objetivo de este proyecto era realizar un sistema de gestión de espacios y eventos culturales. Uno de los aspectos a los que se ha dado mayor importancia a la hora de desarrollar el sistema es la facilidad de uso y la amigabilidad de la interfaz de usuario, así como la creación de una arquitectura de desarrollo que facilitase la continuación de la aplicación en futuras evoluciones.

Otro punto relevante, aunque quizá pase desapercibido para el usuario final, ha sido la gestión de la concurrencia a los datos almacenados en el sistema. Para esto se han investigado y utilizado mecanismos de control de bloqueos, transparentes para el usuario final, que permitirán que la información esté disponible para todos los usuarios sin exponer la integridad de los datos almacenados.

Se puede observar que las necesidades principales siempre han sido muy claras y concretas, gracias a la posibilidad de haber trabajado con funcionarios y personas encargadas de la gestión cultural como parte de su actividad diaria, las cuales han proporcionado bastantes ideas y correcciones sobre el desarrollo a ejecutar.

Durante la redacción de la memoria siempre se ha tenido muy presente la evolución del proyecto y de las distintas partes que componen el sistema, haciendo que aunque el desarrollo del mismo puede parecer sencillo en su presentación, en muchos casos la complejidad de algunos elementos se ha visto incrementada notablemente para conseguir que estos fueran fácilmente mantenibles y evolucionables en un futuro, como por ejemplo el sistema de creación de detalles técnicos de los espacios.

Sobre los objetivos establecidos al comienzo del proyecto, podemos decir que se han alcanzado todos ellos:

- El sistema permitirá administrar y controlar los usuarios que van a emplear la aplicación y organizarlos por tantos grupos de trabajo como quieran definirse dentro de la organización y



compartir toda la información común.

- Permitirá gestionar espacios y eventos e incrementar la cantidad de información almacenada para cada uno de ellos según se precise, así como realizar el seguimiento de los eventos programados de forma fácil e intuitiva.
- Se ha encontrado una solución que satisface la problemática relativa a la diversidad de datos existentes entre los distintos e incluso los mismos tipos de espacios culturales que será capaz de gestionar la aplicación.
- El sistema está preparado para admitir el multiidioma, aunque para los efectos de este proyecto de fin de carrera sólo se haya utilizado el castellano.
- El sistema y la arquitectura diseñada permitirán futuras evoluciones y mejoras de la aplicación incluida la integración con nuevos sistemas que quieran acceder a la información por él gestionada.

A la hora de hablar de los problemas encontrados para la realización del proyecto, debemos hablar de la gran variedad de datos técnicos que se necesitan manejar a la hora de almacenar información relativa a espacios culturales y a la complejidad de los mismos para alguien que no tenga experiencia en su gestión y trabaje con ello día a día. Afortunadamente, se ha podido contar con el apoyo de diverso personal encargado de la gestión de espacios culturales en distintas organizaciones, que han podido asesorar, corregir y probar distintos aspectos de la aplicación durante el desarrollo de la misma.

Para finalizar, es obligatorio mencionar que el tiempo empleado para la elaboración de este proyecto de fin de carrera, debido a diversas circunstancias, sobre todo profesionales, se ha visto afectado por largos periodos de falta de dedicación, lo que ha implicado que su desarrollo se incrementase hasta aproximadamente un año de trabajo.

5.2 Futuros trabajos

Desde el inicio del proyecto se tuvo en consideración que no se incluirían en la aplicación todos los requisitos funcionales necesarios para crear un sistema que cubriese completamente las necesidades de los usuarios consultados, ya que debía acotarse el tamaño de la misma en términos de tiempo y de tamaño, dejando las partes ausentes a futuras evoluciones del sistema.

Sin embargo, se han cubierto todos los requisitos principales detectados en el primer análisis del sistema y en las distintas reuniones mantenidas con el personal que ha facilitado la creación de la aplicación.

A continuación se detallan distintos requerimientos a desarrollar o que podría ser interesante desarrollar en futuras evoluciones:



- **Gestión de presupuestos:** Sería interesante para los usuarios el poder mantener una gestión de presupuestos de cada grupo de trabajo. Esto permitiría que, si cada grupo gestionase los presupuestos de que dispone, por ejemplo, en los distintos ejercicios fiscales, se pudiese planificar de antemano los eventos que se pueden contratar en base a dichos presupuestos. Para esto, además de la creación del módulo de gestión de presupuestos, habría que implementar un módulo de contabilidad por grupo, y extender el módulo de gestión de eventos para incluir el importe que supone cada uno de los eventos que se programen.
- **Gestión contable:** Como añadido a la gestión de presupuestos, se podría añadir al sistema un módulo de gestión contable global o por grupo de trabajo, que junto a la gestión de presupuestos permitiría conocer en cada momento el presupuesto de que dispone un grupo para la gestión cultural. Para ello, se podría, o bien desarrollar un módulo de contabilidad al completo o, debido al gran número de herramientas de este tipo disponibles en el mercado, tanto de pago como gratuitas, analizar la posibilidad de integrar el sistema con una de ellas o en su caso con la utilizada con el usuario.
- **Gestión de empleados:** Tal como se indica en uno de los requisitos del sistema, en un futuro se quiere integrar en la aplicación una gestión completa de empleados. En futuros trabajos se deberá analizar detalladamente los datos a almacenar sobre los mismos para poder alcanzar una gestión de personal que satisfaga las necesidades requeridas.
- **Generación de Listados:** Aunque en la mayor parte de los negocios y en los organismos de la administración pública se está instaurando la oficina sin papel, podría ser interesante la opción de poder generar e imprimir los listados mostrados en la aplicación.
- **Importar datos:** Aunque no hay gran cantidad de aplicaciones de gestión de espacios y eventos culturales, es posible que actualmente se esté realizando esta gestión con algún producto propietario o incluso realizándose a mano. Sería interesante permitir importar datos en nuestra aplicación procedente de otros sistemas, siempre siguiendo una estructura determinada.
- **Integración con otros sistemas:** Aunque el sistema actual permite una gestión integral de los espacios y eventos culturales de diversos grupos de trabajo es interesante que esta información, además de ser gestionada, pueda ser expuesta a los usuarios finales o asistentes a dichos espectáculos. Para ello, se podría preparar el sistema para integrarse con otros sistemas externos y que estos pudiesen recoger y exponer la información recogida en el sistema GEEC. Por ejemplo, se podrían querer listar todos los espacios culturales de Leganés en la página web del consistorio, lo cual se podría lograr exponiendo la información mediante un Webservice. Lo mismo ocurriría para los eventos culturales de un grupo.
- **Gestión de explotación de espacios:** Como el sistema ya almacena información sobre aforos de los espacios culturales y sobre los eventos en ellos organizados, se podría añadir un nuevo módulo al sistema que permitiese gestionar la venta de entradas para cada uno de ellos, lo cual



enlazaría perfectamente con los módulos de contabilidad y presupuestos.

- **Gestión del personal de los espacios:** Aunque se ha incluido un módulo de gestión de personal, también sería interesante añadir la gestión de personal de los distintos espacios culturales, ya que esta información también puede ser interesante a la hora de los requisitos necesarios para la asignación de eventos.

De cualquier forma, estos son los puntos que personalmente considero más importantes o atractivos para futuras evoluciones, tras haber trabajado con posibles usuarios del sistema, ya que, como en todo desarrollo, las posibilidades de ampliación de la aplicación pueden ser muy amplias.



6 ANEXO A. Scripts de base de datos

6.1 – Script de creación de base de datos**6.6 – Script de inicialización de la tabla estado****6.2 – Script inicialización de la tabla de actividad****6.7 – Error: No se encuentra la fuente de referencia****6.3 – Script inicialización de la tabla de comunidad****6.8 – Script de inicialización de la tabla tipo_entidad****6.4 – Script de inicialización de la tabla provincia****6.9 – Script de inicialización de la tabla tipo_persona****6.5 – Script de inicialización de la tabla municipio**

Este anexo contiene los scripts de DDL (Data Definition Language) y de DML (Data Manipulation Language) de base de datos para la creación del esquema de base de datos de la aplicación, y las sentencias de inserción de los datos necesarios para la inicialización de la misma.

6.1 Script de creación de base de datos

```
/* ===== */
/* Sentencia de borrado de tablas del esquema. */
/* ===== */
DROP TABLE `usuario_rol`;
DROP TABLE `usuario_rol_grupo`;
DROP TABLE `empleado`;
DROP TABLE `contacto`;
DROP TABLE `persona`;
DROP TABLE `tipo_persona`;
DROP TABLE `usuario`;
DROP TABLE `rol`;
DROP TABLE `evento`;
DROP TABLE `estado`;
DROP TABLE `empresa`;
DROP TABLE `actividad`;
DROP TABLE `espacio`;
DROP TABLE `entidad`;
DROP TABLE `municipio`;
DROP TABLE `provincia`;
DROP TABLE `comunidad_autonoma`;
DROP TABLE `tipo_entidad`;
DROP TABLE `grupo`;

/* ===== */
/* TABLA: usuario */
/* DESCRIPCIÓN: Sentencia de creación de la tabla de usuarios del sistema. */
```



```
/* ===== */
CREATE TABLE `usuario` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(50) NOT NULL,
  `clave` VARCHAR(50) NOT NULL,
  `habilitado` TINYINT(1) DEFAULT '1' NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `idx_nombreUsuario_unique` UNIQUE (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: rol
/* DESCRIPCIÓN: Sentencia de creación de la tabla de roles de usuarios.
/* ===== */
CREATE TABLE `rol` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `idx_nombreRol_unique` UNIQUE (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: grupo
/* DESCRIPCIÓN: Sentencia de creación de la tabla de usuarios del sistema.
/* ===== */
CREATE TABLE `grupo` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `idx_nombreGrupo_unique` UNIQUE (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: usuario_rol
/* DESCRIPCIÓN: Sentencia de creación de la tabla de usuarios y roles
/* asignados. Se podrá borrar o actualizar usuarios, actualizando los datos
/* de esta tabla en cascada, pero no se podrán borrar roles mientras existan
/* usuarios con dicho rol asignado. Sí se podrá actualizar rol, actualizando
/* en cascada los datos de esta tabla.
/* ===== */
CREATE TABLE `usuario_rol` (
  `idUserio` BIGINT(20) NOT NULL,
  `idRol` BIGINT(20) NOT NULL,
  PRIMARY KEY (`idUserio`, `idRol`),
  CONSTRAINT `fk_usuario_usuarioRol`
    FOREIGN KEY (`idUserio`) REFERENCES `usuario` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_rol_usuarioRol`
    FOREIGN KEY (`idRol`) REFERENCES `rol` (`id`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: usuario_rol_grupo
/* DESCRIPCIÓN: Sentencia de creación de la tabla de usuarios, roles y grupos
/* a los que pertenecen. Se podrán borrar usuarios, borrando los registros de
/* esta tabla en cascada, pero no se podrán borrar grupos ni roles mientras
/* existan usuarios asignados a ellos. Se podrán actualizar los usuarios,
/* roles y grupos, actualizando en cascada los datos de esta tabla.
/* ===== */
CREATE TABLE `usuario_rol_grupo` (
  `idUserio` BIGINT(20) NOT NULL,
  `idRol` BIGINT(20) NOT NULL,
  `idGrupo` BIGINT(20) NOT NULL,
  PRIMARY KEY (`idUserio`, `idRol`, `idGrupo`),
  CONSTRAINT `fk_usuario_usuarioRolGrupo`
    FOREIGN KEY (`idUserio`) REFERENCES `usuario` (`id`)
    ON DELETE CASCADE
```



```
        ON UPDATE CASCADE,
    CONSTRAINT `fk_rol_usuarioRolGrupo`
        FOREIGN KEY (`idRol`) REFERENCES `rol` (`id`)
        ON DELETE RESTRICT
        ON UPDATE CASCADE,
    CONSTRAINT `fk_grupo_usuarioRolGrupo`
        FOREIGN KEY (`idGrupo`) REFERENCES `grupo` (`id`)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: comunidad_autonoma */
/* DESCRIPCIÓN: Sentencia de creación de la tabla de comunidades autónomas. */
/* El nombre de la comunidad autónoma debe ser único. */
/* ===== */
CREATE TABLE `comunidad_autonoma` (
    `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
    `nombre` VARCHAR(50) NOT NULL,
    PRIMARY KEY (`id`),
    CONSTRAINT `idx_nombreComunidad_unique` UNIQUE (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: provincia */
/* DESCRIPCIÓN: Sentencia de creación de la tabla de provincias. No se podrá */
/* eliminar una comunidad autónoma mientras tenga provincias asignadas a ella */
/* pero si se actualiza, se actualizarán automáticamente todas sus */
/* provincias. Además el nombre de la provincia debe ser único. */
/* ===== */
CREATE TABLE `provincia` (
    `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
    `nombre` VARCHAR(50) NOT NULL,
    `idComunidad` BIGINT(20) NOT NULL,
    PRIMARY KEY (`id`),
    CONSTRAINT `fk_comunidad_provincia`
        FOREIGN KEY (`idComunidad`) REFERENCES `comunidad_autonoma` (`id`)
        ON DELETE RESTRICT
        ON UPDATE CASCADE,
    CONSTRAINT `idx_nombreProvincia_unique` UNIQUE (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: municipio */
/* DESCRIPCIÓN: Sentencia de creación de la tabla de municipios. No se podrá */
/* eliminar una provincia mientras tenga municipios asignadas a ella, pero si */
/* se actualiza, se actualizarán automáticamente todos sus municipios. */
/* ===== */
CREATE TABLE `municipio` (
    `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
    `nombre` VARCHAR(50) NOT NULL,
    `idProvincia` BIGINT(20) NOT NULL,
    PRIMARY KEY (`id`),
    CONSTRAINT `fk_provincia_municipio`
        FOREIGN KEY (`idProvincia`) REFERENCES `provincia` (`id`)
        ON DELETE CASCADE
        ON UPDATE RESTRICT
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: tipo_entidad */
/* DESCRIPCIÓN: Sentencia de creación de la tabla de tipos de entidad. Aunque */
/* el tipo de entidad es el discriminante utilizado en la tabla 'entidad' */
/* para distinguir entre los subtipos, se ha separado en una tabla a parte */
/* lo que permitirá adaptar la base de datos a futuros subtipos. El nombre de */
/* tipo de entidad debe ser único. */
/* ===== */
CREATE TABLE `tipo_entidad` (
    `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
    `tipo` VARCHAR(50) NOT NULL,
```




```
PRIMARY KEY (`id`),
UNIQUE KEY `idx_tipoEntidad_unique` (`tipo`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: entidad */
/* DESCRIPCIÓN: Sentencia de creación de la tabla de entidades. No se podrá */
/* eliminar ni actualizar un tipo de entidad mientras exista alguna entidad */
/* que utilice ese tipo. Tampoco se podrán eliminar comunidades, provincias */
/* o municipios siempre que pertenezcan a alguna entidad, aunque es */
/* improbable que se de el caso al ser tablas clave de la aplicación. */
/* ===== */
CREATE TABLE `entidad` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(100) NOT NULL,
  `direccion` VARCHAR(200),
  `codigoPostal` VARCHAR(5),
  `idComunidad` BIGINT(20),
  `idProvincia` BIGINT(20),
  `idMunicipio` BIGINT(20),
  `telefono` VARCHAR(9),
  `fax` VARCHAR(11),
  `correoElectronico` VARCHAR(50),
  `web` VARCHAR(100),
  `idTipo` BIGINT(20) NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `fk_tipoEntidad_entidad`
    FOREIGN KEY (`idTipo`) REFERENCES `tipo_entidad` (`id`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT,
  CONSTRAINT `fk_comunidad_entidad`
    FOREIGN KEY (`idComunidad`) REFERENCES `comunidad_autonoma` (`id`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT,
  CONSTRAINT `fk_provincia_entidad`
    FOREIGN KEY (`idProvincia`) REFERENCES `provincia` (`id`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT,
  CONSTRAINT `fk_municipio_entidad`
    FOREIGN KEY (`idMunicipio`) REFERENCES `municipio` (`id`)
    ON DELETE RESTRICT
    ON UPDATE RESTRICT
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: actividad */
/* DESCRIPCIÓN: Sentencia de creación de la tabla de actividades de empresas. */
/* Se ha creado una tabla de actividades para poder gestionar las mismas. El */
/* nombre de la actividad será único en la tabla. */
/* ===== */
CREATE TABLE `actividad` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `idx_nombreActividad_unique` UNIQUE (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: empresa */
/* DESCRIPCIÓN: Sentencia de creación de la tabla de empresas. Si se elimina */
/* o se actualiza una entidad se eliminará o actualizará también su registro */
/* de la tabla de empresas. No se podrá eliminar una actividad de empresa si */
/* existe alguna empresa que la utilice, pero si se podrá actualizar, */
/* actualizando en cascada todas las empresas que la utilicen. Además, el CIF */
/* de la empresa debe ser único. */
/* ===== */
CREATE TABLE `empresa` (
  `idEntidad` BIGINT(20) NOT NULL,
  `idActividad` BIGINT(20) NOT NULL,
  `cif` VARCHAR(9),
```



```
`cuentaBancaria` VARCHAR(20),
CONSTRAINT `idx_cif_unique` UNIQUE (`cif`),
/* CONSTRAINT `idx_cuentaBancaria_unique` UNIQUE (`cuentaBancaria`), */
CONSTRAINT `fk_entidad_empresa`
    FOREIGN KEY (`idEntidad`) REFERENCES `entidad` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
CONSTRAINT `fk_actividad_empresa`
    FOREIGN KEY (`idActividad`) REFERENCES `actividad` (`id`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: tipo_persona */
/* DESCRIPCIÓN: Sentencia de creación de la tabla de tipos de persona. Aunque */
/* el tipo de persona es el discriminante utilizado en la tabla 'persona' */
/* para distinguir entre los subtipos, se ha separado en una tabla a parte */
/* lo que permitirá adaptar la base de datos a futuros subtipos. Además, el */
/* tipo de persona no podrá repetirse dentro de la tabla. */
/* ===== */
CREATE TABLE `tipo_persona` (
    `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
    `tipo` VARCHAR(50) NOT NULL,
    PRIMARY KEY (`id`),
    UNIQUE KEY `idx_tipoPersona_unique` (`tipo`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: persona */
/* DESCRIPCIÓN: Sentencia de creación de la tabla de personas. No se podrá */
/* eliminar ni actualizar un tipo de persona mientras exista alguna persona */
/* que utilice ese tipo. Además el DNI de una persona deberá ser único. */
/* ===== */
CREATE TABLE `persona` (
    `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
    `nombre` VARCHAR(50) NOT NULL,
    `primerApellido` VARCHAR(50) NOT NULL,
    `segundoApellido` VARCHAR(50) NOT NULL,
    `nif` VARCHAR(9) NOT NULL,
    `idTipo` BIGINT(20) NOT NULL,
    PRIMARY KEY (`id`),
    UNIQUE KEY `idx_dni_unique` (`nif`),
    CONSTRAINT `fk_tipoPersona_persona`
        FOREIGN KEY (`idTipo`) REFERENCES `tipo_persona` (`id`)
        ON DELETE RESTRICT
        ON UPDATE RESTRICT
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: empleado */
/* DESCRIPCIÓN: Sentencia de creación de la tabla de empleados. Si se elimina */
/* o se actualiza una persona se eliminará o actualizará también su registro */
/* de la tabla de empleado. Si se elimina el usuario del empleado, se pondrá */
/* a null el valor del identificador de empleado de esta tabla. Además el */
/* identificador de usuario será único, es decir, una persona sólo puede */
/* tener un usuario y un usuario sólo podrá pertenecer a una persona. */
/* ===== */
CREATE TABLE `empleado` (
    `idPersona` BIGINT(20) NOT NULL,
    `horarioIrregular` TINYINT(1) NOT NULL DEFAULT '0',
    `horasConvenio` DECIMAL(4,2) NOT NULL DEFAULT '40.00',
    `idUsuario` BIGINT(20) DEFAULT NULL,
    UNIQUE KEY `idx_usuario_unique` (`idUsuario`),
    CONSTRAINT `fk_persona_empleado`
        FOREIGN KEY (`idPersona`) REFERENCES `persona` (`id`)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    CONSTRAINT `fk_usuario_empleado`
        FOREIGN KEY (`idUsuario`) REFERENCES `usuario` (`id`)
```



```
ON DELETE SET NULL
ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: contacto
/* DESCRIPCIÓN: Sentencia de creación de la tabla de contactos. Si se elimina
/* o se actualiza una persona se eliminará o actualizará también su registro
/* de la tabla de contacto. No se podrá eliminar una empresa mientras tenga
/* algún contacto asignado a la misma, aunque sí se podrá actualizar.
/* ===== */
CREATE TABLE `contacto` (
  `idPersona` BIGINT(20) NOT NULL,
  `puesto` VARCHAR(50) NOT NULL,
  `departamento` VARCHAR(50),
  `telefonoPrincipal` VARCHAR(9) NOT NULL,
  `telefonoSecundario` VARCHAR(9),
  `telefonoMovil` VARCHAR(9),
  `fax` VARCHAR(11),
  `correoElectronico` VARCHAR(50),
  `idEmpresa` BIGINT(20) NOT NULL,
  CONSTRAINT `fk_persona_contacto`
    FOREIGN KEY (`idPersona`) REFERENCES `persona` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_empresa_contacto`
    FOREIGN KEY (`idEmpresa`) REFERENCES `empresa` (`idEntidad`)
    ON DELETE CASCADE
    ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: espacio
/* DESCRIPCIÓN: Sentencia de creación de la tabla de espacios. Si se elimina
/* o se actualiza una entidad se eliminará o actualizará también su registro
/* de la tabla de espacios. No se podrá eliminar un grupo si existe algún
/* espacio que lo utilice, pero sí se podrá actualizar, actualizando en
/* cascada todos los espacios que lo utilicen.
/* ===== */
CREATE TABLE `espacio` (
  `idEntidad` BIGINT(20) NOT NULL,
  `detallesTecnicos` TEXT,
  `habilitado` TINYINT(1) DEFAULT '1' NOT NULL,
  `idGrupo` BIGINT(20) NOT NULL,
  CONSTRAINT `fk_entidad_espacio`
    FOREIGN KEY (`idEntidad`) REFERENCES `entidad` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_grupo_espacio`
    FOREIGN KEY (`idGrupo`) REFERENCES `grupo` (`id`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: estado
/* DESCRIPCIÓN: Sentencia de creación de la tabla de estados de eventos. Se
/* ha creado una tabla de estados para poder gestionar los mismos en el caso
/* de que se quiera ampliar el número de estados. El nombre del estado será
/* único en la tabla.
/* ===== */
CREATE TABLE `estado` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(50) NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `idx_nombreEstado_unique` UNIQUE (`nombre`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: evento
```



```
/* DESCRIPCIÓN: Sentencia de creación de la tabla de eventos. No se podrá */
/* eliminar un estado de evento mientras que algún evento se encuentre en */
/* dicho estado, pero sí se podrá actualizar modificando en cascada todos los */
/* eventos que se encuentren en dicho estado. No se podrán eliminar espacios */
/* ni empresas relacionadas con el evento, pero sí se podrán actualizar en */
/* cascada, modificando el o los eventos que las incluyan. */
/* ===== */
CREATE TABLE `evento` (
  `id` BIGINT(20) NOT NULL AUTO_INCREMENT,
  `nombre` VARCHAR(150) NOT NULL,
  `descripcion` TEXT,
  `fechaInicio` DATETIME NOT NULL,
  `fechaFin` DATETIME NOT NULL,
  `fechaModificacion` DATETIME,
  `idEstado` BIGINT(20) NOT NULL,
  `idEspacio` BIGINT(20) NOT NULL,
  `idEmpresa` BIGINT(20) NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `fk_estado_evento`
    FOREIGN KEY (`idEstado`) REFERENCES `estado` (`id`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
  CONSTRAINT `fk_espacio_evento`
    FOREIGN KEY (`idEspacio`) REFERENCES `espacio` (`idEntidad`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
  CONSTRAINT `fk_empresa_evento`
    FOREIGN KEY (`idEmpresa`) REFERENCES `empresa` (`idEntidad`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;

/* ===== */
/* TABLA: evento_historico */
/* DESCRIPCIÓN: Sentencia de creación de la tabla del histórico de eventos. */
/* No se podrán eliminar espacios ni empresas relacionadas con el evento que */
/* se encuentre en esta tabla, pero sí se podrán actualizar en cascada, */
/* modificando el o los eventos que las incluyan. */
/* ===== */
CREATE TABLE `evento_historico` (
  `id` BIGINT(20) NOT NULL,
  `nombre` VARCHAR(150) NOT NULL,
  `descripcion` TEXT,
  `fechaInicio` DATETIME NOT NULL,
  `fechaFin` DATETIME NOT NULL,
  `fechaModificacion` DATETIME,
  `idEstado` BIGINT(20) NOT NULL,
  `idEspacio` BIGINT(20) NOT NULL,
  `idEmpresa` BIGINT(20) NOT NULL,
  PRIMARY KEY (`id`),
  CONSTRAINT `fk_estado_eventoHistorico`
    FOREIGN KEY (`idEstado`) REFERENCES `estado` (`id`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
  CONSTRAINT `fk_espacio_eventoHistorico`
    FOREIGN KEY (`idEspacio`) REFERENCES `espacio` (`idEntidad`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE,
  CONSTRAINT `fk_empresa_eventoHistorico`
    FOREIGN KEY (`idEmpresa`) REFERENCES `empresa` (`idEntidad`)
    ON DELETE RESTRICT
    ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```



6.2 Script de inicialización de la tabla de usuario

Es obligatorio que la tabla de usuarios contenga al menos al superusuario para que pueda configurar la aplicación.

```
insert into `usuario` (`id`, `nombre`, `clave`, `habilitado`) values (1, 'admin', '123', 1);
```

6.3 Script de inicialización de la tabla rol

```
insert into `rol` (`id`, `nombre`) values (1, 'Superadministrador');  
insert into `rol` (`id`, `nombre`) values (2, 'Administrador');  
insert into `rol` (`id`, `nombre`) values (3, 'Programador');  
insert into `rol` (`id`, `nombre`) values (4, 'Autorizador');
```

6.4 Script inicialización de la tabla de usuario_rol

Es obligatorio que la tabla de roles de usuario contenga al menos un registro asociando al usuario superadministrador el rol de superadministrador, para que pueda acceder a la aplicación sin pertenecer a ningún grupo y así configurar los datos necesarios.

```
insert into `usuario_rol` (`idUserio`, `idRol`) values (1, 1);
```

6.5 Script inicialización de la tabla de actividad

```
insert into `actividad` (`id`, `nombre`) values (1, 'Teatro');  
insert into `actividad` (`id`, `nombre`) values (2, 'Teatro infantil');  
insert into `actividad` (`id`, `nombre`) values (3, 'Música');  
insert into `actividad` (`id`, `nombre`) values (4, 'Música infantil');  
insert into `actividad` (`id`, `nombre`) values (5, 'Ballet clásico');
```

6.6 Script inicialización de la tabla de comunidad

```
insert into `comunidad_autonoma` (`id`, `nombre`) values (1, 'Andalucía');  
insert into `comunidad_autonoma` (`id`, `nombre`) values (2, 'Aragón');  
insert into `comunidad_autonoma` (`id`, `nombre`) values (3, 'Canarias');  
insert into `comunidad_autonoma` (`id`, `nombre`) values (4, 'Cantabria');  
insert into `comunidad_autonoma` (`id`, `nombre`) values (6, 'Castilla la mancha');  
insert into `comunidad_autonoma` (`id`, `nombre`) values (5, 'Castilla y León');
```



```
insert into `comunidad_autonoma` (`id`, `nombre`) values (7, 'Cataluña');
insert into `comunidad_autonoma` (`id`, `nombre`) values (8, 'Comunidad de Madrid');
insert into `comunidad_autonoma` (`id`, `nombre`) values (9, 'Comunidad foral de Navarra');
insert into `comunidad_autonoma` (`id`, `nombre`) values (10, 'Comunidad Valenciana');
insert into `comunidad_autonoma` (`id`, `nombre`) values (11, 'Extremadura');
insert into `comunidad_autonoma` (`id`, `nombre`) values (12, 'Galicia');
insert into `comunidad_autonoma` (`id`, `nombre`) values (13, 'Islas Baleares');
insert into `comunidad_autonoma` (`id`, `nombre`) values (14, 'La Rioja');
insert into `comunidad_autonoma` (`id`, `nombre`) values (15, 'País Vasco');
insert into `comunidad_autonoma` (`id`, `nombre`) values (16, 'Principado de Asturias');
insert into `comunidad_autonoma` (`id`, `nombre`) values (17, 'Región de Murcia');
insert into `comunidad_autonoma` (`id`, `nombre`) values (18, 'Ciudad autónoma de Ceuta');
insert into `comunidad_autonoma` (`id`, `nombre`) values (19, 'Ciudad autónoma de Melilla');
```

6.7 Script de inicialización de la tabla provincia

```
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (1, 'Almería', 1);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (2, 'Cádiz', 1);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (3, 'Córdoba', 1);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (4, 'Granada', 1);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (5, 'Huelva', 1);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (6, 'Jaén', 1);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (7, 'Málaga', 1);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (8, 'Sevilla', 1);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (9, 'Huesca', 2);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (10, 'Teruel', 2);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (11, 'Zaragoza', 2);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (12, 'Las Palmas', 3);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (13, 'Tenerife', 3);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (14, 'Cantabria', 4);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (15, 'Ávila', 5);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (16, 'Burgos', 5);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (17, 'León', 5);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (18, 'Palencia', 5);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (19, 'Salamanca', 5);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (20, 'Segovia', 5);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (21, 'Soria', 5);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (22, 'Valladolid', 5);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (23, 'Zamora', 5);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (24, 'Albacete', 6);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (25, 'Ciudad Real', 6);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (26, 'Cuenca', 6);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (27, 'Guadalajara', 6);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (28, 'Toledo', 6);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (29, 'Barcelona', 7);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (30, 'Gerona', 7);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (31, 'Lérida', 7);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (32, 'Tarragona', 7);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (33, 'Madrid', 8);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (34, 'Navarra', 9);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (35, 'Alicante', 10);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (36, 'Castellón', 10);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (37, 'Valencia', 10);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (38, 'Cáceres', 11);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (39, 'Badajoz', 11);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (40, 'La Coruña', 12);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (41, 'Lugo', 12);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (42, 'Orense', 12);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (43, 'Pontevedra', 12);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (44, 'Islas Baleares', 13);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (45, 'La Rioja', 14);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (46, 'Álava', 15);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (47, 'Guipúzcoa', 15);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (48, 'Vizcaya', 15);
```



```
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (49, 'Asturias', 16);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (50, 'Murcia', 17);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (51, 'Ceuta', 18);
insert into `provincia` (`id`, `nombre`, `idComunidad`) values (52, 'Melilla', 19);
```

6.8 Script de inicialización de la tabla municipio

De este script sólo se incluirá un ejemplo debido a su extensión, ya que el script original utiliza datos reales proporcionados por la administración pública y tiene cerca de 20000 registros.

```
insert into `municipio` (`id`, `nombre`, `idProvincia`) values (1, 'Abejuela', 1);
insert into `municipio` (`id`, `nombre`, `idProvincia`) values (2, 'Abla', 1);
insert into `municipio` (`id`, `nombre`, `idProvincia`) values (3, 'Abrucena', 1);
insert into `municipio` (`id`, `nombre`, `idProvincia`) values (4, 'Adra', 1);
insert into `municipio` (`id`, `nombre`, `idProvincia`) values (5, 'Agua Amarga', 1);
insert into `municipio` (`id`, `nombre`, `idProvincia`) values (6, 'Agua De Medio', 1);
insert into `municipio` (`id`, `nombre`, `idProvincia`) values (7, 'Aguadulce', 1);
insert into `municipio` (`id`, `nombre`, `idProvincia`) values (8, 'Albanchez', 1);
insert into `municipio` (`id`, `nombre`, `idProvincia`) values (9, 'Alboloduy', 1);
insert into `municipio` (`id`, `nombre`, `idProvincia`) values (10, 'Albox', 1);
insert into `municipio` (`id`, `nombre`, `idProvincia`) values (11, 'Alcaudique', 1);
```

6.9 Script de inicialización de la tabla estado

Este script contiene los estados que inicialmente admitirá un evento y por los que irá pasando según se vayan confirmando aspectos de su ciclo de vida.

```
INSERT INTO `estado` (`id`, `nombre`) VALUES (1, 'Pendiente');
INSERT INTO `estado` (`id`, `nombre`) VALUES (2, 'Aceptado');
INSERT INTO `estado` (`id`, `nombre`) VALUES (3, 'Rechazado');
INSERT INTO `estado` (`id`, `nombre`) VALUES (4, 'Ejecutado');
```

6.10 Script de inicialización de la tabla tipo_entidad

En un principio los tipos de entidad serán estáticos, pero al mantenerlos en una tabla hacemos que el sistema sea extensible de una forma sencilla.

```
insert into `tipo_entidad` (`id`, `tipo`) values (1, 'Espacio');
insert into `tipo_entidad` (`id`, `tipo`) values (2, 'Empresa');
```



6.11 Script de inicialización de la tabla tipo_persona

En un principio los tipos de persona serán estáticos, pero al mantenerlos en una tabla hacemos que el sistema sea extensible de una forma sencilla.

```
insert into `tipo_persona` (`id`, `tipo`) values (1, 'Empleado');  
insert into `tipo_persona` (`id`, `tipo`) values (2, 'Contacto');
```




7 ANEXO B. Manual de instalación

Este manual va dirigido al usuario encargado de la instalación y configuración del sistema para su posterior puesta en marcha y explotación.

Debido a las características técnicas del lenguaje de programación elegido para el desarrollo de la aplicación, el sistema podrá ejecutarse igualmente sobre cualquier sistema Windows, Linux, etc..., siempre y cuando se respeten los requisitos mínimos de software para que la misma funcione, los cuales se enumeran a continuación:

- Sistemas operativos soportados: Windows XP, Windows Vista, Windows 7 (con .NET Framework 4 instalado), Linux en cualquiera de las distribuciones que disponga de paquetes de instalación para MySQL y Apache Tomcat.
- Máquina virtual Java JRE 6 o JDK 6.
- Base de datos MySQL v5 o superior, aunque puede reconfigurarse la aplicación para trabajar con cualquier otro tipo de base de datos como Oracle, MSSQL Server, etc...
- Contenedor servlet Apache Tomcat 6, aunque también puede utilizarse un servidor de aplicaciones JEE ya sea libre o de pago como GlassFish, TomEE, WebLogic, etc...
- Para la explotación de la aplicación, los usuarios deberán tener instalado un navegador web como Internet Explorer, Mozilla Firefox, Google Chrome, Opera, etc... ya que las páginas desarrolladas para la aplicación se muestran correctamente en cualquiera de ellos.

Debido a estas características multi-plataforma de la aplicación, la redacción de un manual de instalación para todas las plataformas admitidas excede la longitud admitida para este trabajo de fin de carrera, por lo que nos limitaremos a realizar el manual de instalación para el Sistema Operativo Windows, ya que al utilizar software libre para todas las aplicaciones necesarias para el uso del sistema, se puede encontrar bastante información para la instalación de las mismas en Internet.

7.1 Instalación de la Máquina Virtual Java 6

Para la instalación de la máquina virtual Java, accederemos a la dirección de descarga de la misma, disponible en el sitio web de Oracle bajo la URL:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

y seleccionamos la pestaña de descargas, mostrándose la pantalla que se muestra en la Figura 7.1:

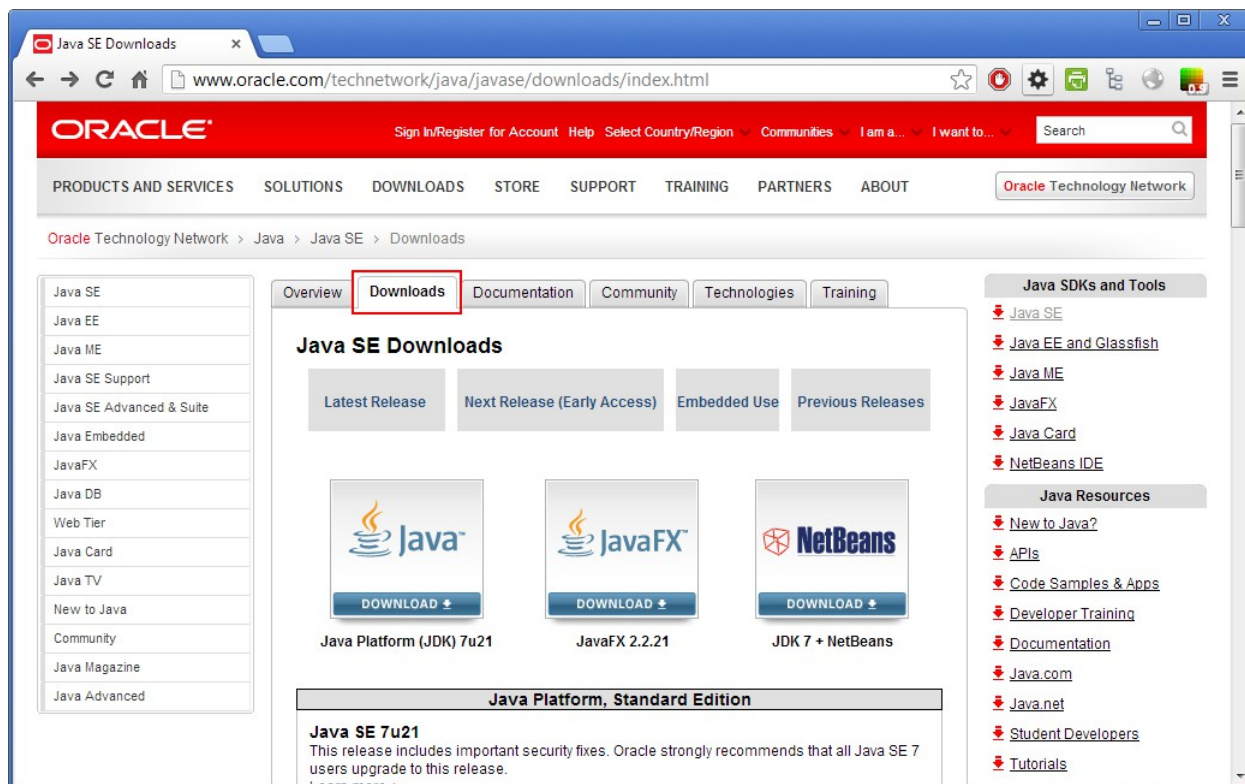


Figura 7.1. Página de descarga de la Java SE de Oracle

Nos desplazamos hacia abajo en la ventana hasta que encontremos el área de descarga de la Java SE JRE 6 y pulsamos el botón **Download** (Figura 7.2):

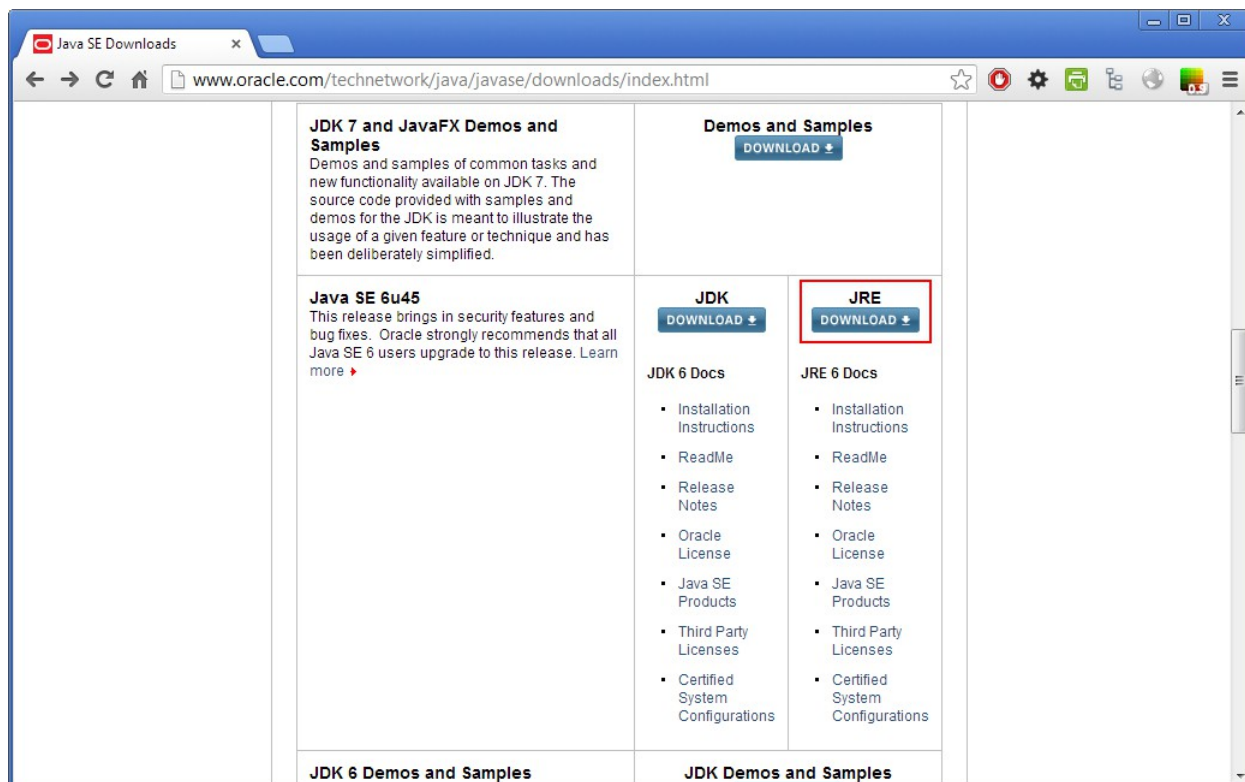


Figura 7.2. Área de descarga de la Java SE 6

En la siguiente ventana del proceso de descarga (Figura 7.3), aceptaremos el acuerdo de licencia (1) y seleccionaremos la plataforma para la cual queremos realizar la descarga. En el ejemplo de este manual hemos seleccionado Windows 32bit y el fichero de instalación offline (2):

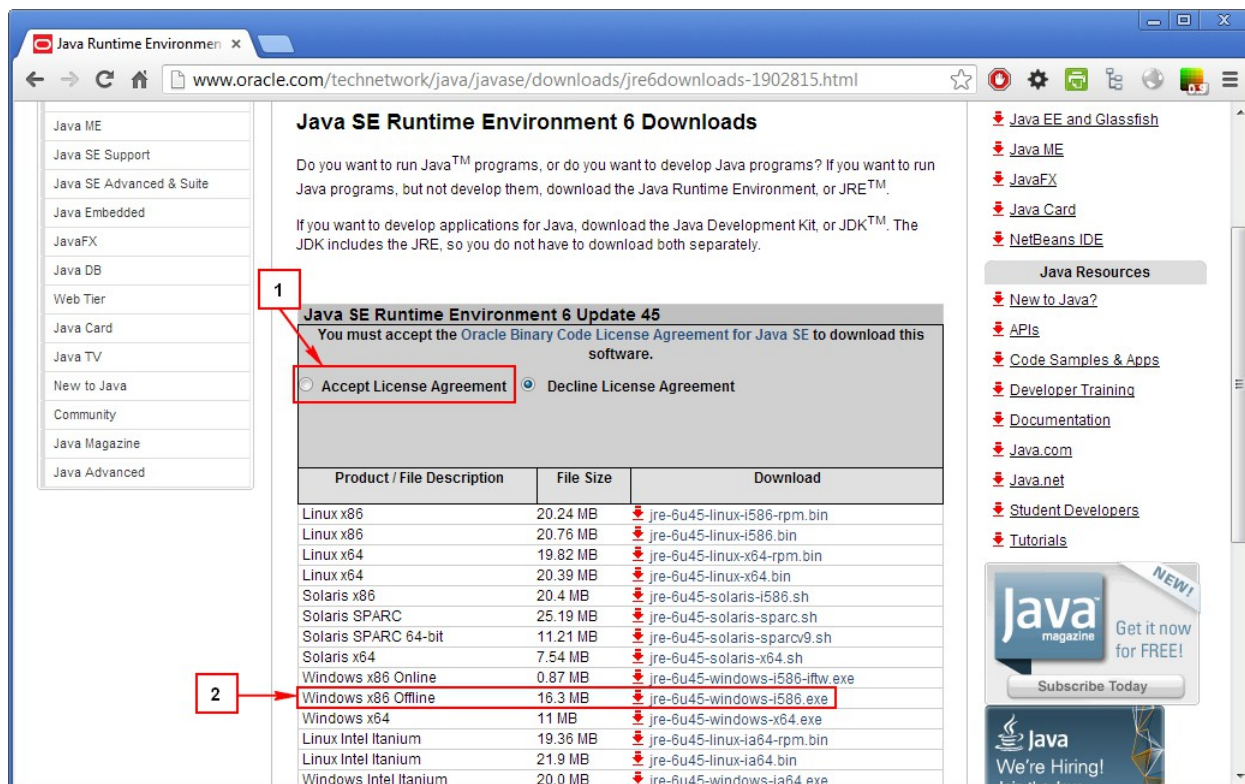


Figura 7.3. Acuerdo de licencia y producto a descargar

Una vez seleccionado el fichero, comenzará la descarga del mismo a la ubicación que hayamos seleccionado, y una vez finalizada la cual podremos comenzar con el proceso de instalación propiamente dicho.

Accedemos a la carpeta donde hayamos descargado el archivo y hacemos doble clic sobre el mismo para iniciar el proceso de instalación.

En la primera ventana del asistente de instalación de la JRE (Figura 7.4) pulsaremos el botón **Instalar** dejando el resto de opciones por defecto, ya que vamos a realizar la instalación en el directorio por defecto:



Figura 7.4. Pantalla inicial de la instalación

Una vez pulsado el botón comenzará el proceso de instalación propiamente dicho, el cual durará varios minutos (en función de la configuración hardware de la máquina en la que se esté instalando), una vez finalizado el cual se mostrará, si todo ha ido correctamente, como podemos ver en la Figura 7.5:



Figura 7.5. Pantalla final de la instalación

Pulsamos el botón **Cerrar** y ya tendremos instalada la máquina virtual Java y configuradas las variables de entorno necesarias para su correcta utilización.

Para comprobar que se ha instalado correctamente, podemos abrir una consola de comandos y ejecutar el siguiente comando de sistema: **java -version**, el cual nos debería proporcionar información sobre la máquina virtual instalada y su versión, tal como podemos ver en la Figura 7.6:



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrador>java -version
java version "1.6.0_45"
Java(TM) SE Runtime Environment (build 1.6.0_45-b06)
Java HotSpot(TM) Client VM (build 20.45-b01, mixed mode, sharing)

C:\Documents and Settings\Administrador>
```

Figura 7.6. Comprobación de la instalación en el sistema

7.2 Instalación del SGBD MySQL 5

El siguiente paso consiste en la instalación del sistema gestor de base de datos MySQL 5 con el cual trabajará el sistema. Para su descarga accederemos a la dirección disponible para tal efecto en el sitio web de MySQL bajo la URL:

<http://dev.mysql.com/downloads/>

y seleccionamos el enlace de descarga con el texto Download, tal como podemos ver en la Figura 7.7.



Figura 7.7. Área de descarga de MySQL



A continuación se mostrará la ventana de descarga en la que, desplazándonos hacia abajo, se nos solicitará seleccionar la plataforma para la cual realizar la descarga del instalador (1). Una vez seleccionada la plataforma, en nuestro caso Microsoft Windows 32 bit, pulsamos el botón **Download** (Figura 7.8).

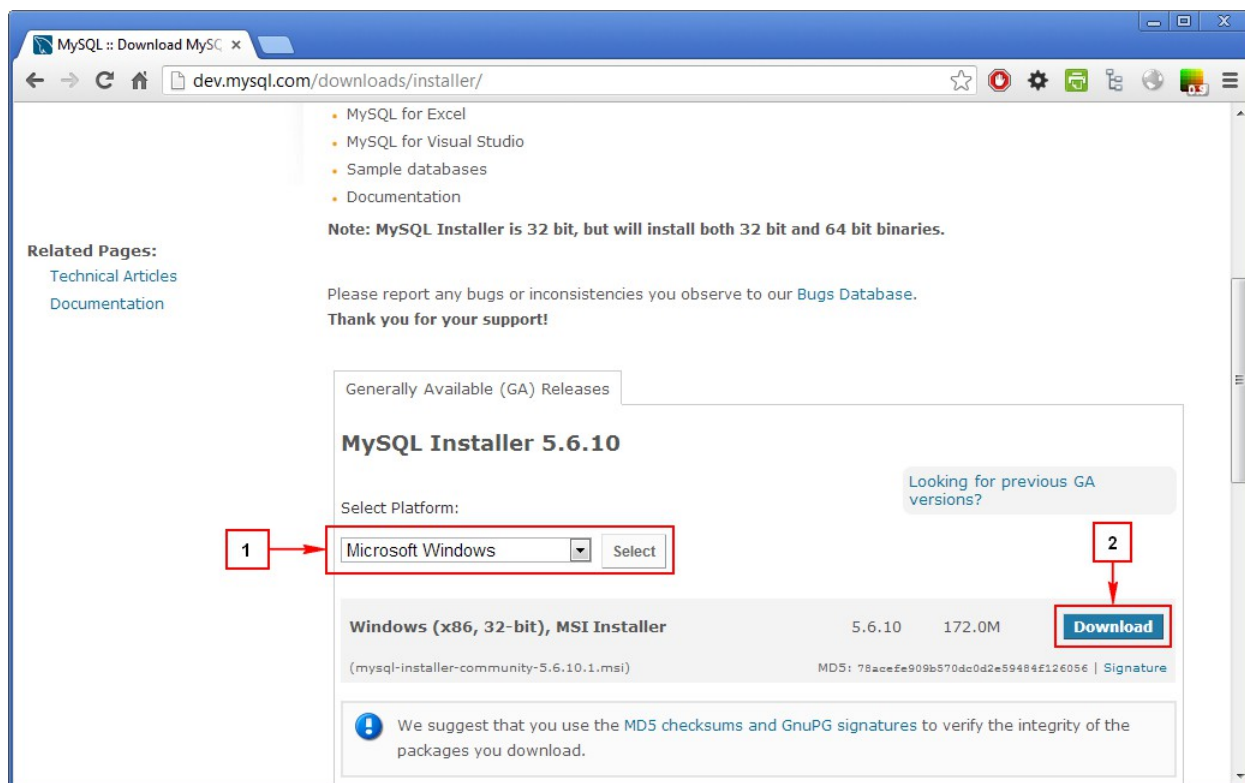


Figura 7.8. Área de descarga y selección de plataforma de MySQL

El proceso de descarga nos lleva a una ventana en la que se nos dará la posibilidad de registrarnos o seleccionar una descarga anónima. Elegiremos esta última opción pulsando en el enlace **No thanks, just start my download** (1), el cual iniciará el proceso de descarga (Figura 7.9).

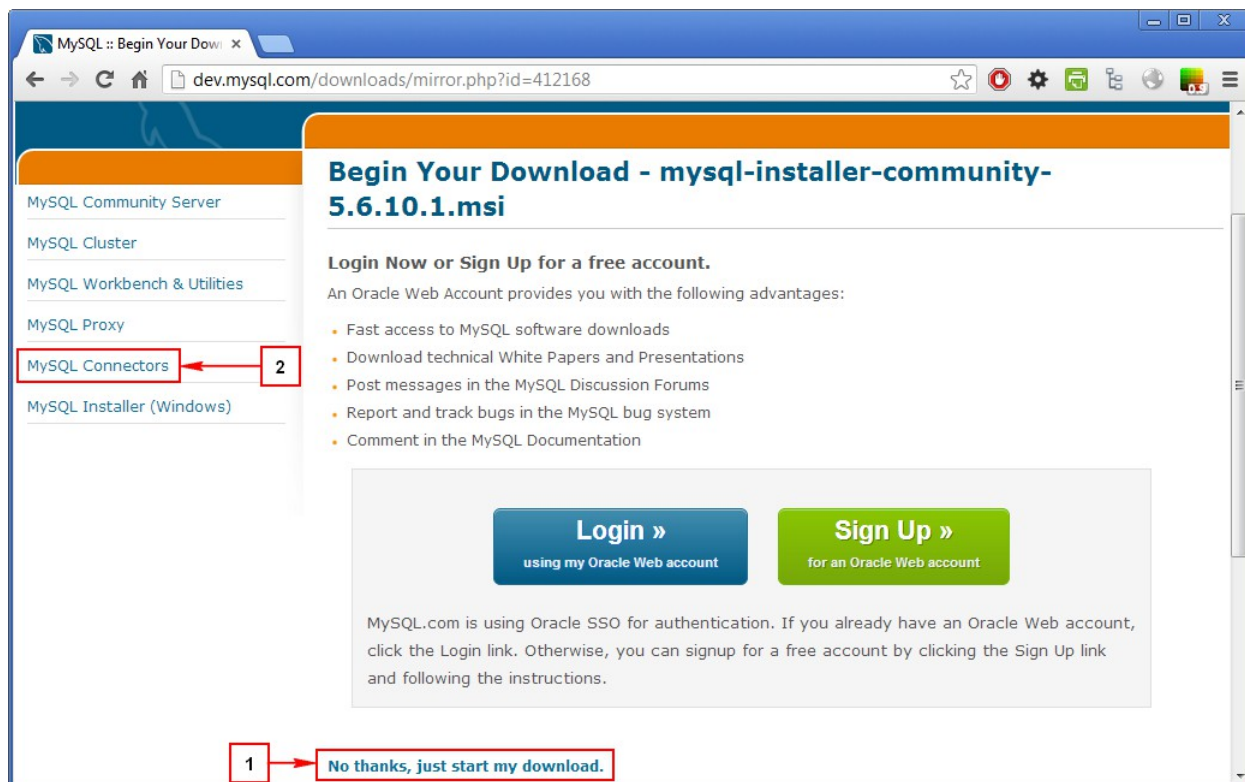


Figura 7.9. Selección del método de descarga, registrado o anónimo

Una vez finalizada la descarga, la cual llevará varios minutos en función de la conexión disponible, procederemos a descargar también el conector Java para la base de datos MySQL que utilizaremos más adelante. Para ello, en esta última ventana pulsaremos en el enlace **MySQL Connectors** que se muestra en el lateral (2). Accederemos a la página de descarga de los distintos conectores MySQL en función del lenguaje de programación que queramos utilizar, la cual podemos ver en la Figura 7.10. En nuestro caso tendremos que seleccionar **ConnectorJ**, que es el utilizado en desarrollos Java.

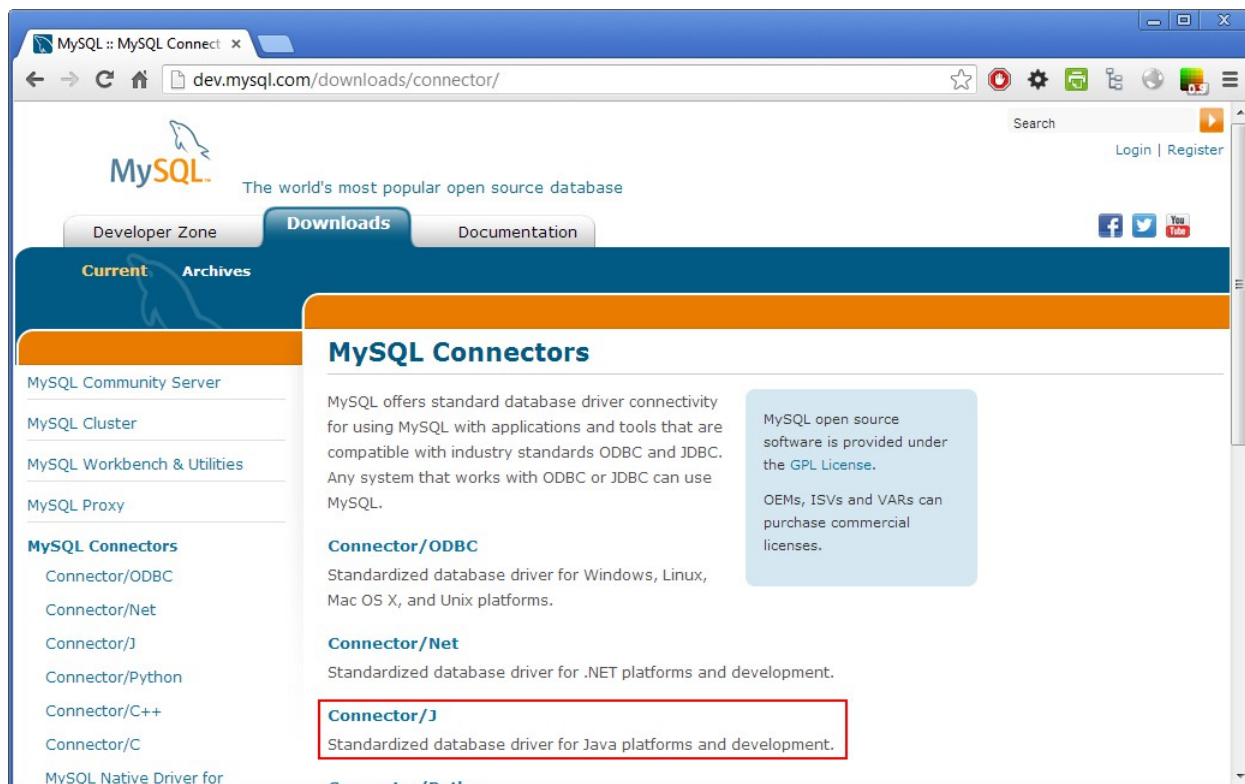


Figura 7.10. Página de descarga de conectores MySQL

Pulsamos el enlace y accedemos a la página de descarga, en la cual nos desplazaremos hacia la parte inferior de la misma y seleccionaremos el formato de descarga más adecuado para nuestras necesidades, en nuestro caso en formato ZIP (Figura 7.11). Pulsamos el botón **Download** y, al igual que en el proceso de descarga del SGBD, se nos pedirá seleccionar la descarga con usuario registrado o la descarga anónima, donde volveremos a seleccionar la misma opción de descarga anónima.

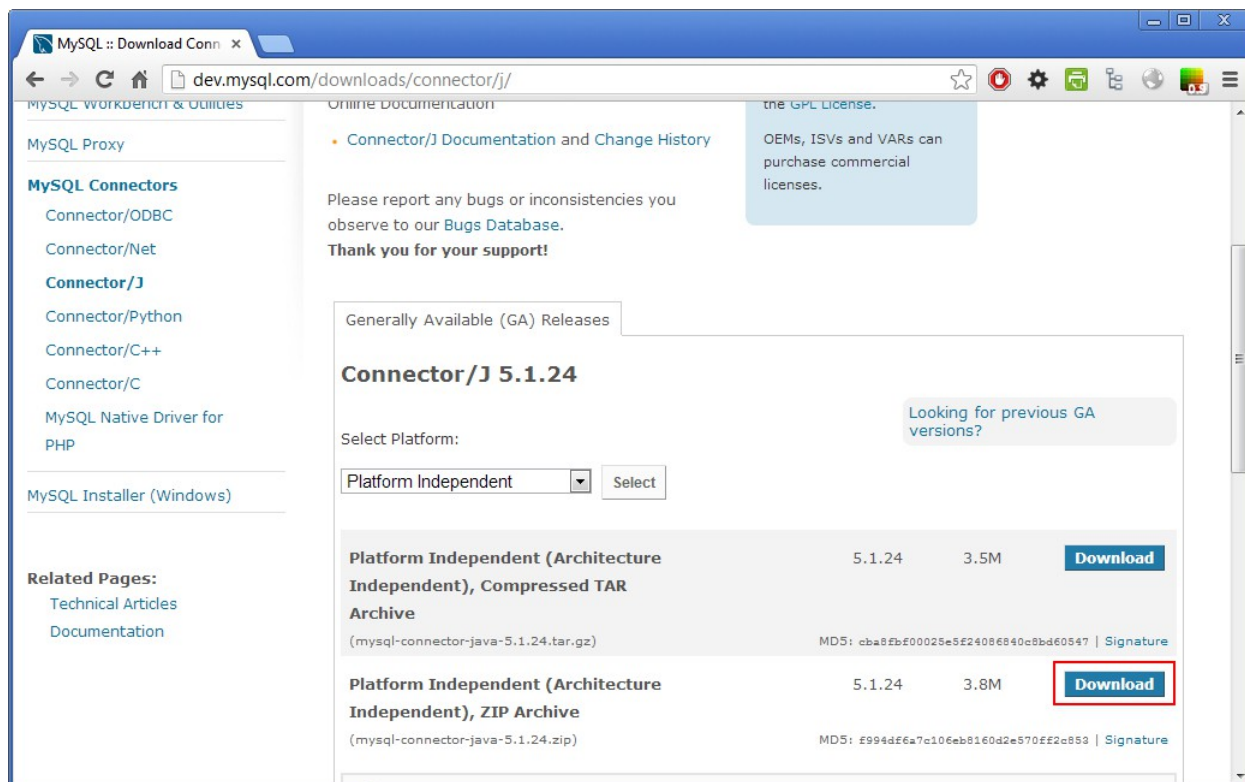


Figura 7.11. Área de descarga, selección de plataforma y formato de descarga del conector

Una vez descargado el instalador del SGBD procederemos a su instalación. Para ello hacemos doble clic en el fichero descargado en la primera parte de este apartado para iniciar la instalación.

La primera pantalla que se muestra es una pantalla de bienvenida, en la cual tendremos que seleccionar la opción **Install MySQL products**, que podemos ver en la Figura 7.12:

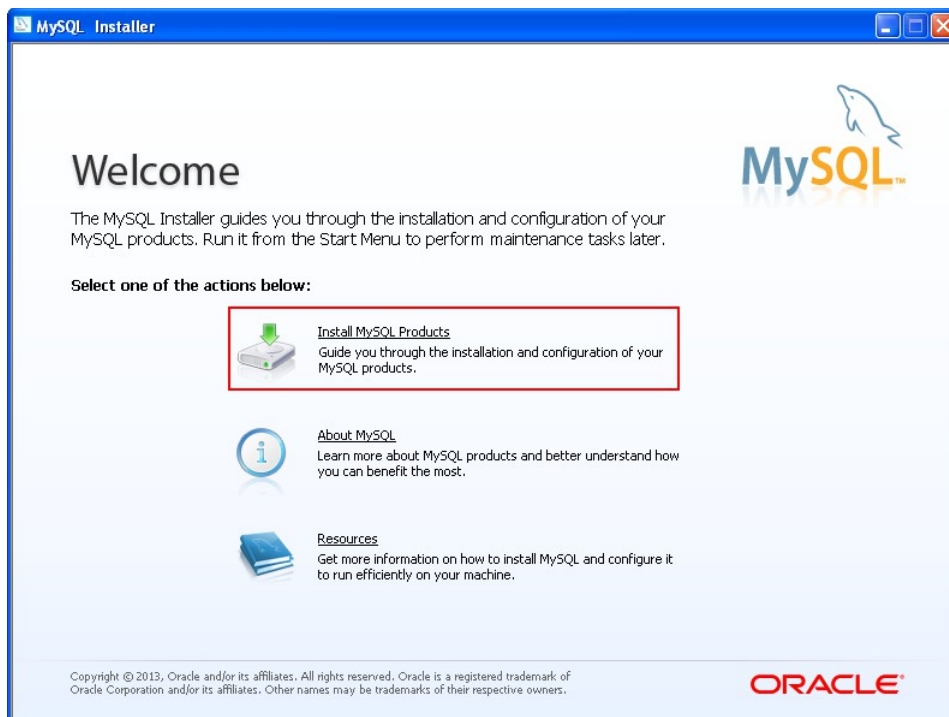


Figura 7.12. Pantalla inicial de instalación de MySQL

La siguiente pantalla del asistente será la del acuerdo de licencia (Figura 7.13), donde seleccionaremos que aceptamos el acuerdo con el checkbox marcado con la etiqueta **I accept the license terms** (1) y pulsaremos el botón **Next** (2) para continuar con el proceso de instalación.

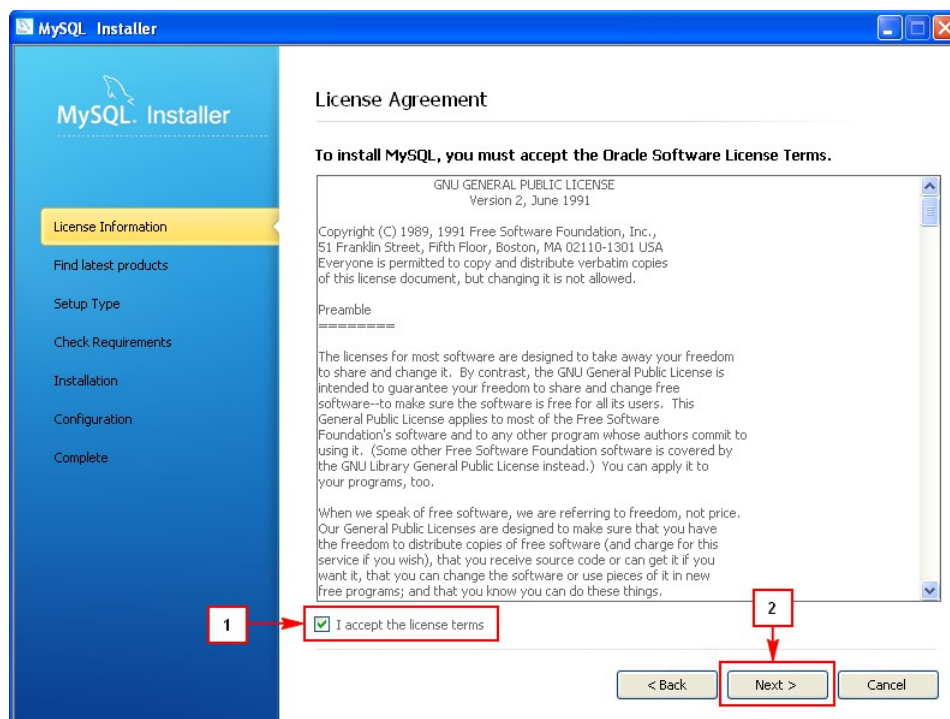


Figura 7.13. Pantalla información del acuerdo de licencia

A continuación se muestra la pantalla en la que se nos pregunta si queremos buscar actualizaciones del producto (Figura 7.14). Seleccionaremos el checkbox marcado con la etiqueta **Skip the check for updates (not recommended)** (1), ya que no queremos buscar actualizaciones y pulsamos el botón **Next** (2).

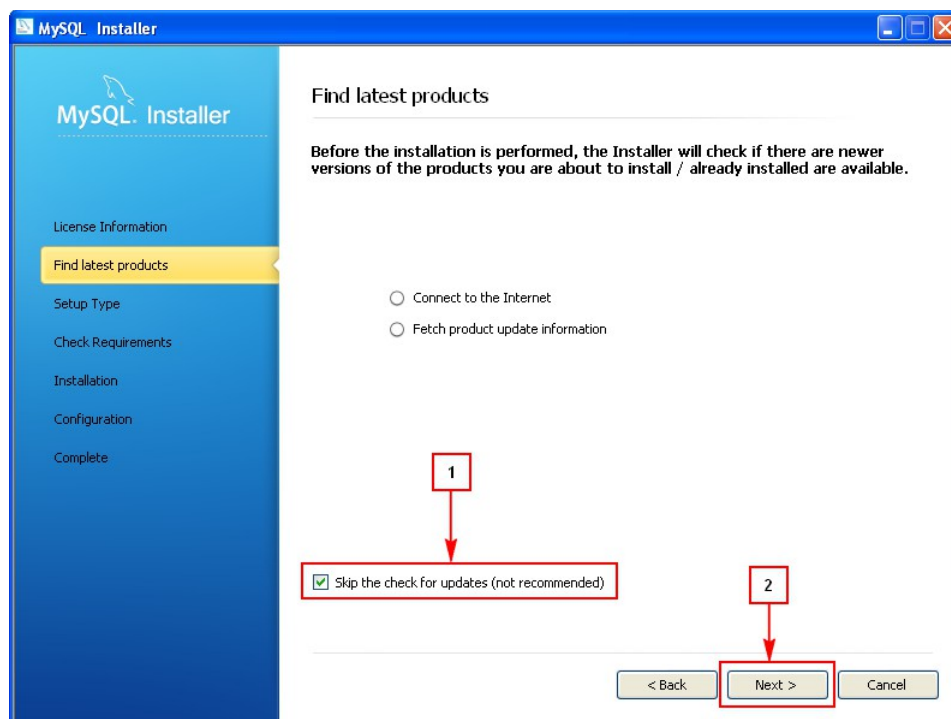


Figura 7.14. Pantalla de selección de búsqueda de actualizaciones

La siguiente pantalla del asistente es la pantalla de selección del tipo de instalación (Figura 7.15). En nuestro caso seleccionaremos la opción **Server only** (1) y pulsaremos el botón **Next** (2).

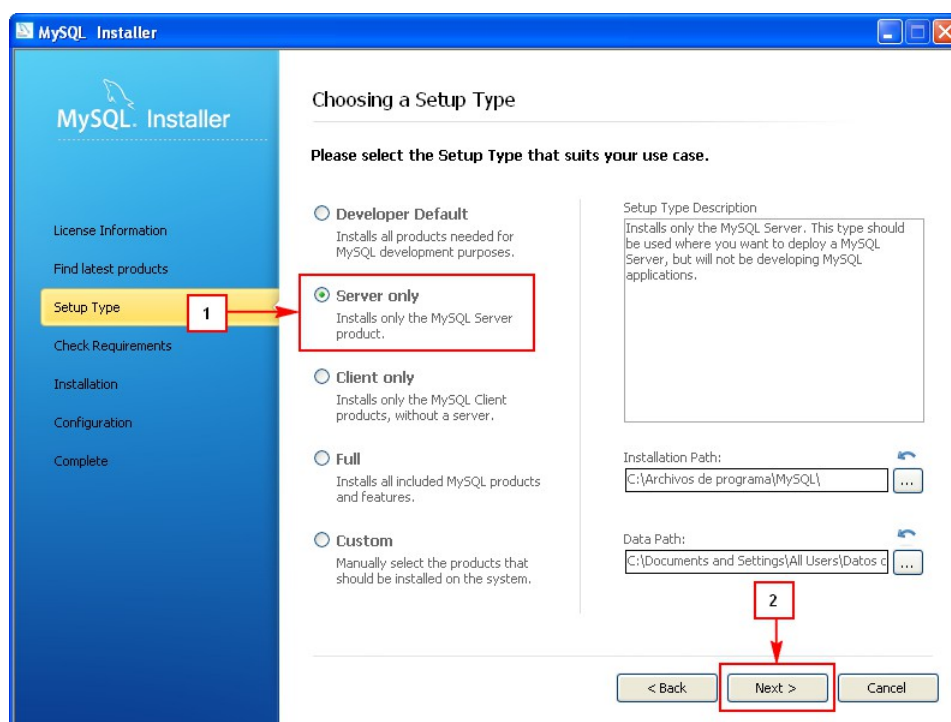


Figura 7.15. Pantalla de selección del tipo de instalación

Se mostrará una pantalla informativa de si la instalación requiere la instalación de componentes adicionales, que no será nuestro caso, y en la que sólo tendremos que pulsar el botón **Next**.

A continuación se mostrará una pantalla informativa de los productos a instalar (Figura 7.16), en la que, tras comprobar que los paquetes a instalar (únicamente el paquete **MySQL Server**) son los correctos pulsaremos el botón **Execute**, tras lo cual comenzará la instalación del producto, mostrando el progreso en esta misma pantalla. Una vez finalizada la instalación, se mostrará en esta pantalla el resultado de la misma. Si todo ha ido correctamente, pulsaremos el botón **Next** para acceder a la pantalla de información de que va a comenzar el proceso de configuración de MySQL.

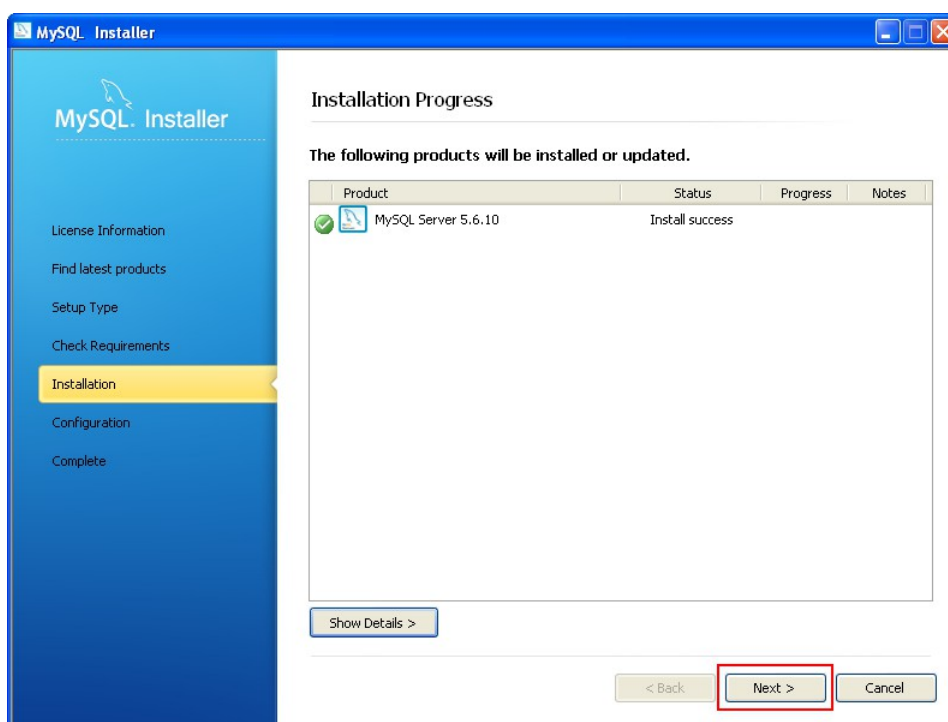


Figura 7.16. Pantalla de información de productos a instalar

En la pantalla de información de comienzo del proceso de configuración (Figura 7.17) sólo tendremos que pulsar el botón **Next**, y comenzará el mismo.

En la primera pantalla de configuración de MySQL se nos pedirá indicar las siguientes características:

- **Server Configuration Type:** Seleccionaremos la opción *Server Machine*, ya que estamos configurando una máquina servidora con opción a que tenga más servidores corriendo en la misma. En este ejemplo, el mismo servidor soportará la instalación del SGBD y del Contenedor Servlet, aunque esto no tendría por qué ser así, y cada uno de ellos podría tener su servidor dedicado.
- **Enable TCP/IP Networking:** Habilitaremos esta opción para permitir las conexiones a base de datos mediante TCP/IP necesarias para nuestra aplicación.

- **Advanced configuration:** Seleccionaremos esta opción para ajustar de forma más precisa la configuración del SGBD.

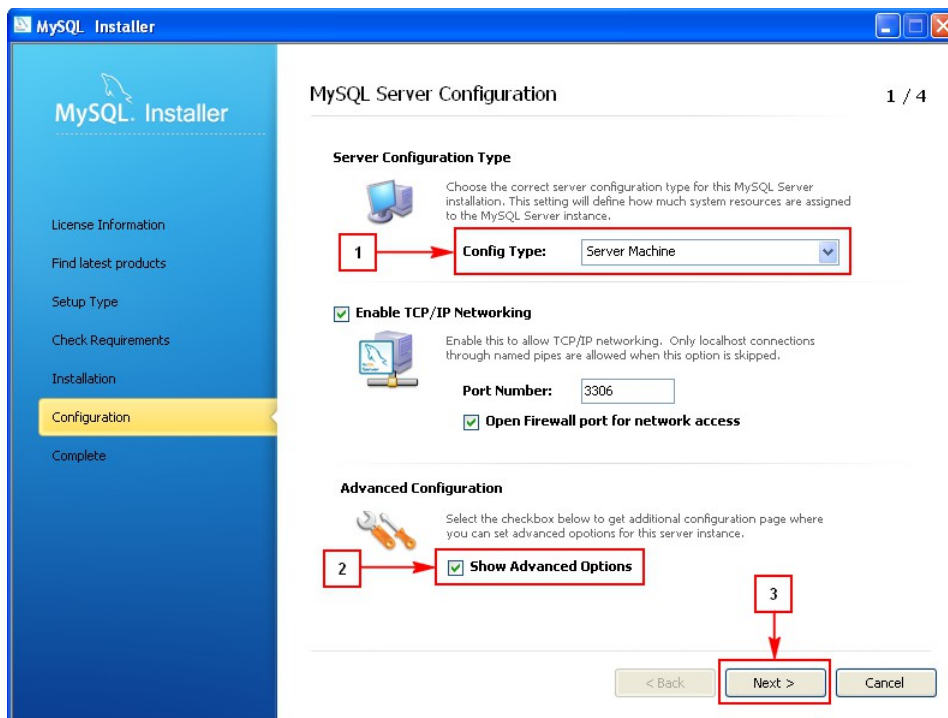


Figura 7.17. Pantalla de configuración de MySQL

Una vez seleccionadas las opciones anteriores, pulsaremos el botón **Next**.

La siguiente pantalla del asistente nos permite configurar la cuenta de administrador o **root**, estableciendo su contraseña, y crear los usuarios de base de datos necesarios (Figura 7.18). Por seguridad, no utilizaremos en nuestra aplicación el usuario administrador, sino que nos crearemos el usuario **geecUser** con la contraseña que deseemos (estos parámetros habrá que configurarlos a posteriori en la aplicación. Una vez configurados los usuarios pulsamos el botón **Next**.

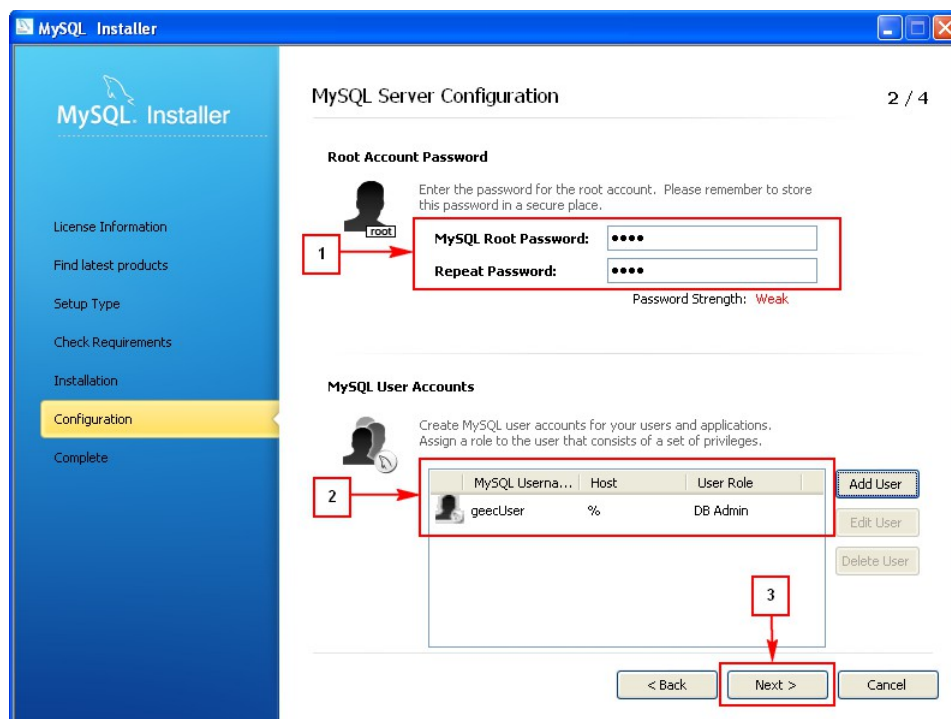


Figura 7.18. Pantalla de configuración de usuarios

La siguiente pantalla nos permite configurar el servicio de Windows que gestionará la instancia del SGBD (Figura 7.19). Dejaremos las opciones por defecto y pulsaremos el botón **Next**.

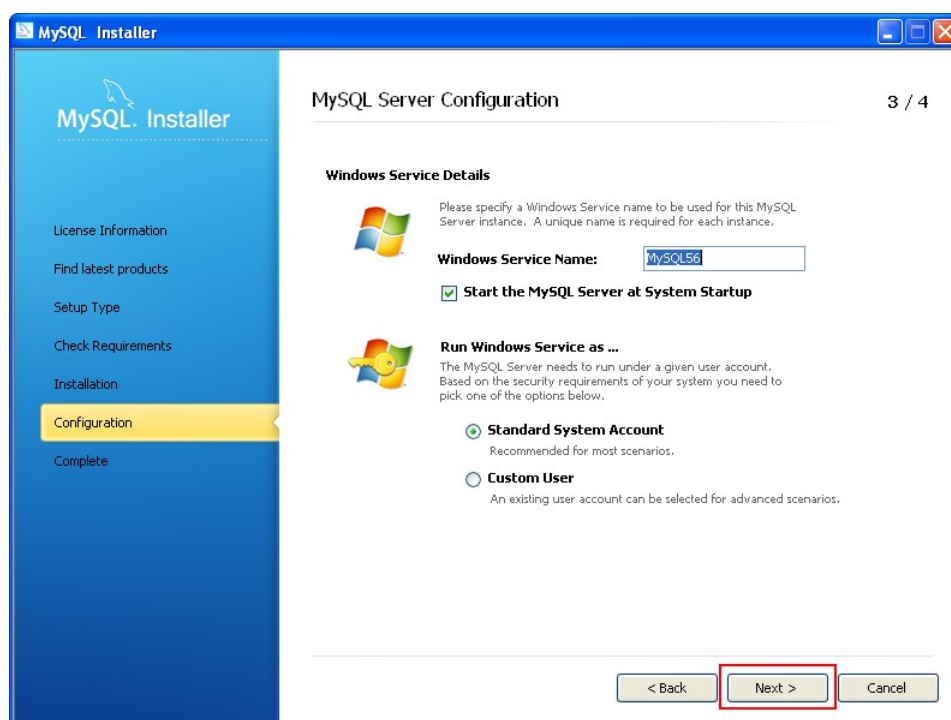


Figura 7.19. Pantalla de configuración del servicio Windows

A continuación podemos configurar los ficheros de Log a utilizar por MySQL (Figura 7.20). En esta pantalla dejaremos también las opciones por defecto y pulsaremos el botón **Next**.

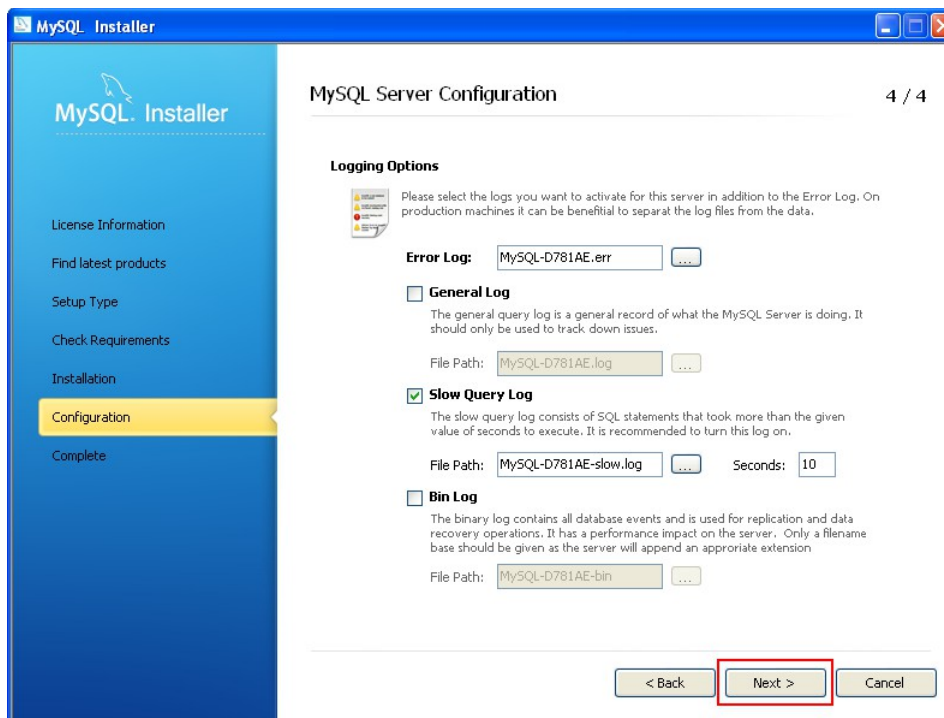


Figura 7.20. Pantalla de configuración del sistema de trazas

Al pulsar el botón Next de la última pantalla de configuración, se procederá al mismo, mostrando una pantalla informativa, en la cual, tras terminar la configuración volveremos a pulsar el botón **Next**, mostrándose una pantalla informativa de finalización del proceso de instalación donde tendremos que pulsar el botón **Finish**.

Una vez hecho esto, ya tendremos instalado y correctamente configurado el SGBD MySQL.

7.3 Creación del esquema de base de datos y ejecución de los scripts de inicialización

Para realizar esta tarea tenemos distintas opciones. Podemos utilizar la interfaz de comandos proporcionada por MySQL o utilizar una herramienta gráfica de administración de bases de datos que facilite la tarea. En nuestro caso nos decantaremos por esta última opción, utilizando software libre al igual que hemos estado haciendo durante todo el proyecto. Para ello utilizaremos **HeidiSQL**.

Para descargar el software **HeidiSQL** accederemos a la dirección de descarga de la misma, disponible en el sitio web de HeidiSQL bajo la URL y que podemos ver en la Figura 7.21:



<http://www.heidisql.com/download.php>

Seleccionaremos el enlace **Installer (Mirror)** de la página e inmediatamente comenzará la descarga del archivo.

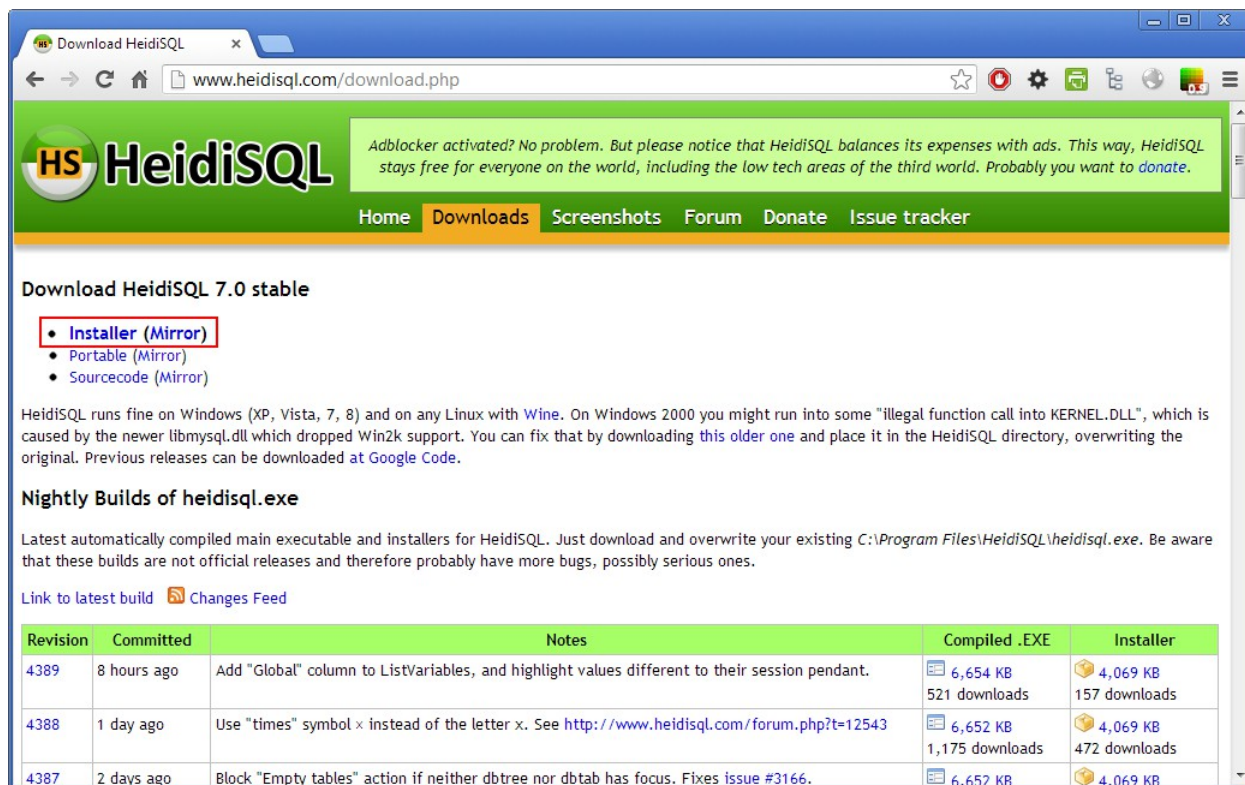


Figura 7.21. Pantalla de descarga de HeidiSQL

Una vez descargado el archivo, haremos doble clic sobre el mismo para iniciar el proceso de instalación.

La primera pantalla del asistente es una pantalla de bienvenida (Figura 7.22), la cual tras pulsar el botón **Next**, nos mostrará la pantalla donde debemos aceptar el acuerdo de licencia. Seleccionaremos la opción **I accept the agreement** y pulsaremos el botón **Next**.

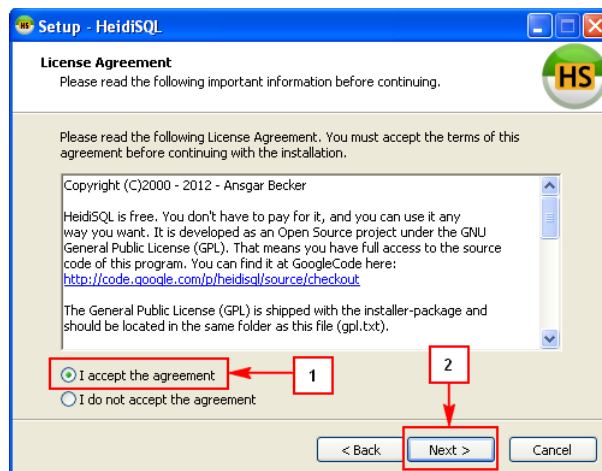


Figura 7.22. Pantalla de información del acuerdo de licencia

A continuación podremos configurar el directorio de instalación del programa (Figura 7.23). Dejaremos el valor por defecto y pulsaremos el botón **Next**.

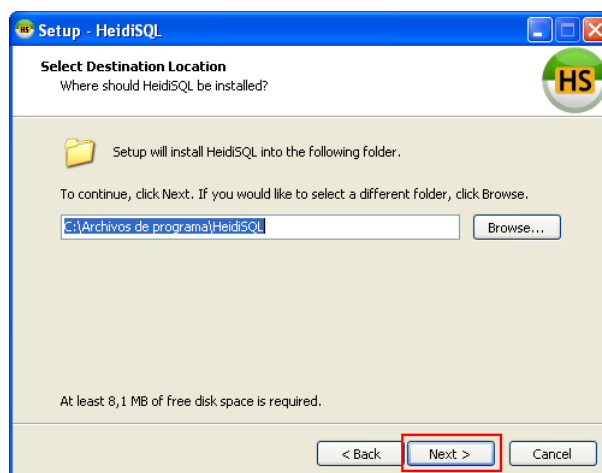


Figura 7.23. Pantalla de selección del directorio de instalación

En la pantalla de creación del grupo de programas, Figura 7.24, dejaremos también el valor por defecto y pulsamos el botón **Next**.

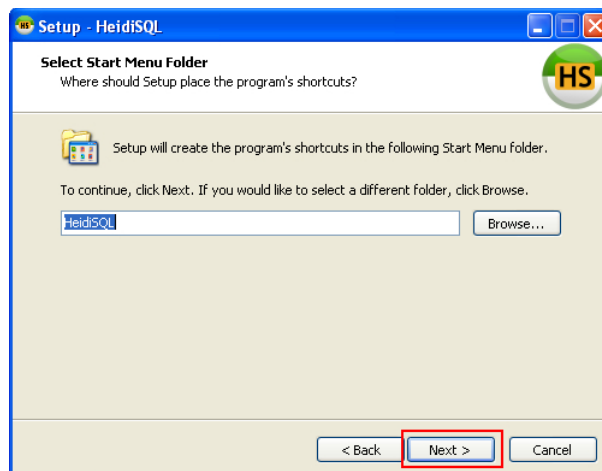


Figura 7.24. Pantalla de configuración del grupo de programas a crear

La siguiente pantalla nos permite configurar la creación de icono de escritorio y las opciones de actualización (Figura 7.25). Seleccionamos las que más nos interesen y pulsamos el botón **Next**. Para el propósito de este manual hemos seleccionado únicamente las opciones **Create a desktop icon** y **Associate .SQL files with HeidiSQL**.

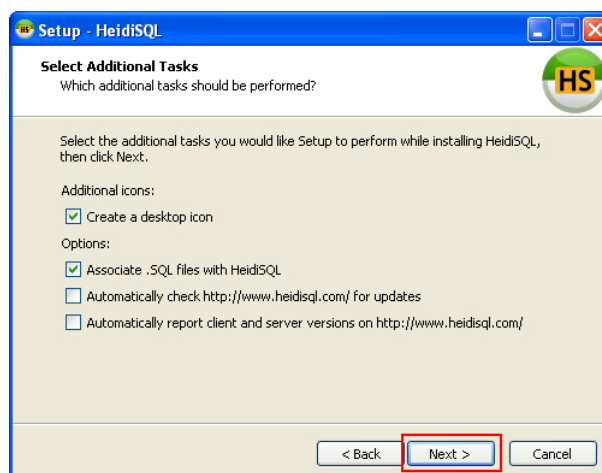


Figura 7.25. Pantalla de opciones de configuración de la instalación

La última pantalla muestra un resumen de las opciones de instalación seleccionadas, que podemos ver en la figura 7.26. Revisaremos las mismas y pulsaremos el botón **Finish** para comenzar la instalación.

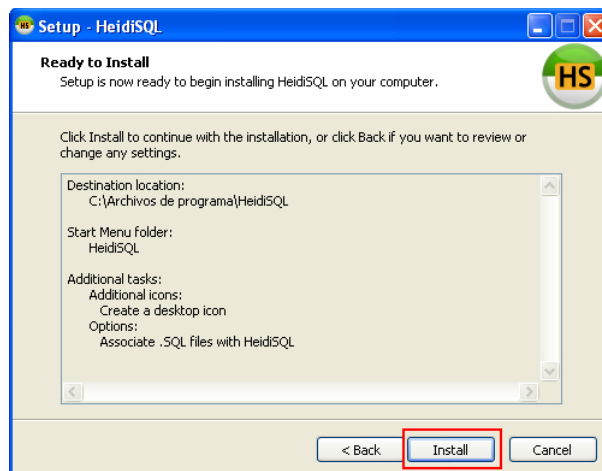


Figura 7.26. Pantalla de información de las opciones de instalación seleccionadas

Una vez finalizada la instalación arrancaremos la aplicación y procederemos a configurar una conexión con el servidor de base de datos para ejecutar los scripts de inicialización del esquema de la aplicación.

Al arrancar el programa se mostrará la pantalla de inicio de sesión en alguno de los esquemas de base de datos que tengamos configurados, tal como podemos observar en la Figura 7.27. En nuestro caso, al ser una instalación nueva no tendremos ninguno, por lo que pasaremos a configurar uno nuevo pulsando el botón **New**.

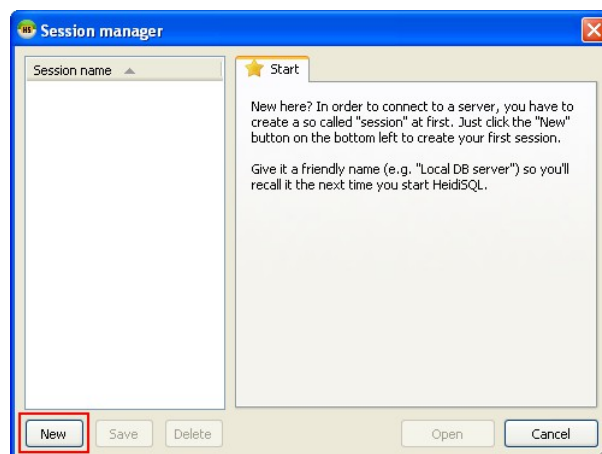


Figura 7.27. Pantalla inicial de HeidiSQL

En esta misma pantalla configuraremos el nombre o alias que queramos darle a la conexión (1), en nuestro caso **geecDB**, el usuario y contraseña a utilizar (2), **geecUser** con la contraseña que hayamos seleccionado al instalar MySQL y por último, grabaremos la conexión y pulsaremos el botón **Open**, tal como podemos observar en la Figura 7.28.

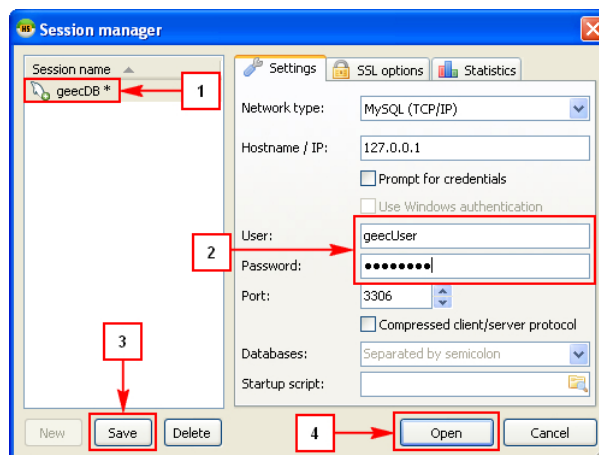


Figura 7.28. Ventana de configuración de una nueva conexión a BBDD.

Se mostrará la pantalla de resumen de los esquemas existentes en el SGBD (si no hemos seleccionado en la pantalla anterior un esquema o **Database** de inicio) que podemos observar en la figura 7.29. Hacemos clic con el botón derecho sobre el nombre de la conexión y seleccionaremos **Create new > Database** para crear el nuevo esquema.

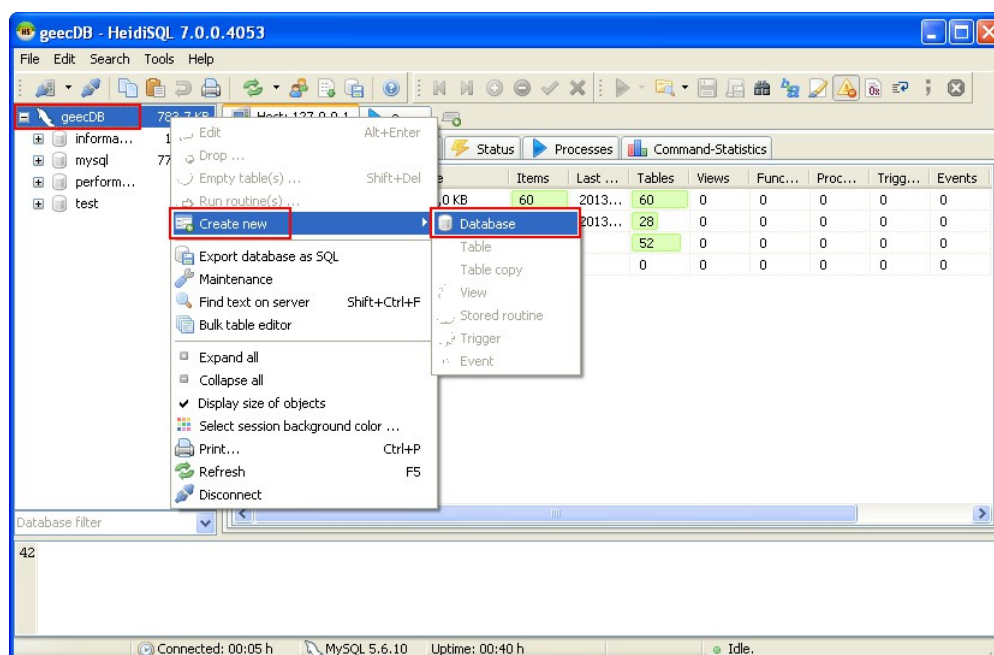


Figura 7.29. Ventana de resumen de la conexión a BBDD y creación de nuevo esquema

En el diálogo que se muestra introduciremos el nombre del esquema a crear, **geecDB** y la codificación a utilizar, la cual dejaremos por defecto en UTF-8 (Figura 7.30). Una vez configurado pulsamos **OK** ya tendremos creado el nuevo esquema.

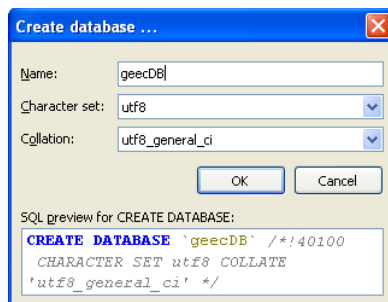


Figura 7.30. Diálogo de creación de esquema

El último paso consistirá en ejecutar los scripts de inicialización incluidos en el anexo A (Figura 7.31). Para ello seleccionaremos el nombre del esquema en la ventana lateral izquierda, y en la parte central accederemos a la pestaña **Query**. Ahí tendremos que pegar el código de las sentencias a ejecutar o importar el fichero **.sql** con las mismas (1) y pulsar el botón **Execute SQL...** (F9) (2).

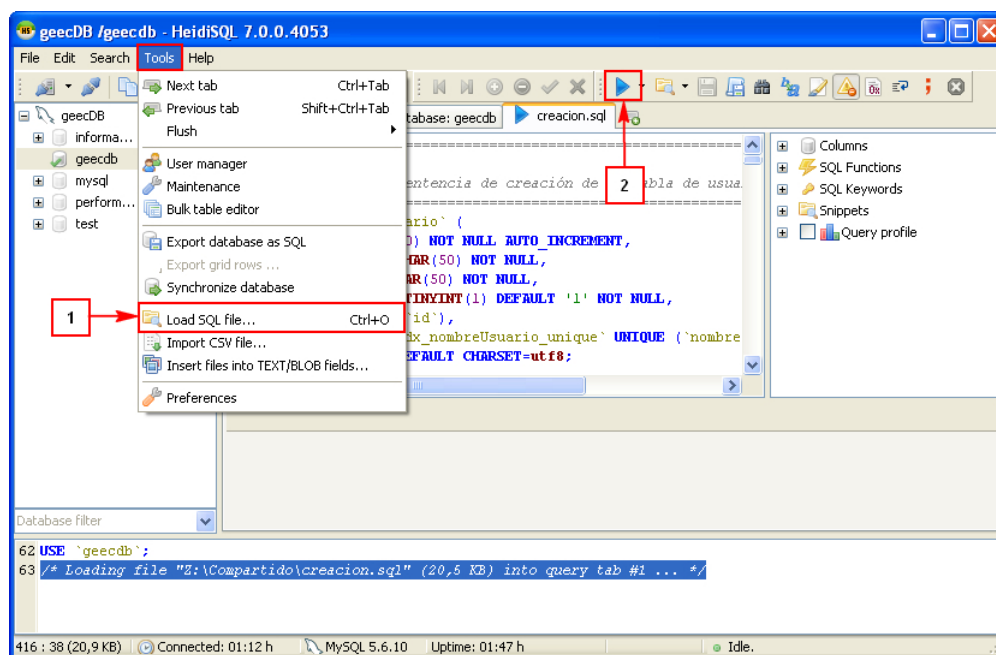


Figura 7.31. Carga de los scripts de inicialización del esquema de base de datos

Una vez finalizada la ejecución de las sentencias (cuyo resultado se nos irá presentando en el log de la parte inferior de la pantalla), por ejemplo las sentencias de creación de las tablas de la base de datos, podremos acceder a la pestaña **Database: geecdb** y consultar el resultado de las mismas (Figura 7.32). Una vez ejecutados todos los scripts, ya tendremos la base de datos inicializada para poder iniciar la aplicación sin ningún problema.



Name	Rows	Size	Created	Updated	Engine	Comment
actividad	0	32,0 KB	2013-04-18 10:52:17		InnoDB	
comunidad_autonoma	0	32,0 KB	2013-04-18 10:52:16		InnoDB	
contacto	0	48,0 KB	2013-04-18 10:52:19		InnoDB	
empleado	0	48,0 KB	2013-04-18 10:52:18		InnoDB	
empresa	0	64,0 KB	2013-04-18 10:52:18		InnoDB	
entidad	0	80,0 KB	2013-04-18 10:52:17		InnoDB	
espacio	0	48,0 KB	2013-04-18 10:52:19		InnoDB	
estado	0	32,0 KB	2013-04-18 10:52:19		InnoDB	
evento	0	64,0 KB	2013-04-18 10:52:19		InnoDB	
evento_historico	0	64,0 KB	2013-04-18 10:52:19		InnoDB	
grupo	0	32,0 KB	2013-04-18 10:52:15		InnoDB	
municipio	0	32,0 KB	2013-04-18 10:52:17		InnoDB	
persona	0	48,0 KB	2013-04-18 10:52:18		InnoDB	
provincia	0	48,0 KB	2013-04-18 10:52:16		InnoDB	
rol	0	32,0 KB	2013-04-18 10:52:15		InnoDB	
tipo_entidad	0	32,0 KB	2013-04-18 10:52:17		InnoDB	

```
88 SHOW FUNCTION STATUS WHERE `Db`='geecdb';
89 SHOW PROCEDURE STATUS WHERE `Db`='geecdb';
90 SHOW TRIGGERS FROM `geecdb`;
91 SHOW EVENTS FROM `geecdb`;
```

Figura 7.32. Pantalla de resumen de las tablas del esquema

7.4 Instalación del Contenedor Servlet Apache Tomcat 6

Para la instalación del Contenedor Servlet Apache Tomcat en el cual desplegaremos el fichero WAR de la aplicación web GEEC, accederemos a la dirección de descarga del mismo, disponible en el sitio web de Apache Tomcat bajo la URL:

<http://tomcat.apache.org/download-60.cgi>

Nos desplazaremos hacia abajo en la página de descarga, hasta la sección de la misma donde se encuentran las distintas distribuciones disponibles (Figura 7.33). En nuestro caso, seleccionaremos el instalador de Tomcat como servicio Windows, ya que este nos permite evitar tener que hacer configuraciones adicionales. Pulsamos el enlace y de inmediato comienza el proceso de descarga.

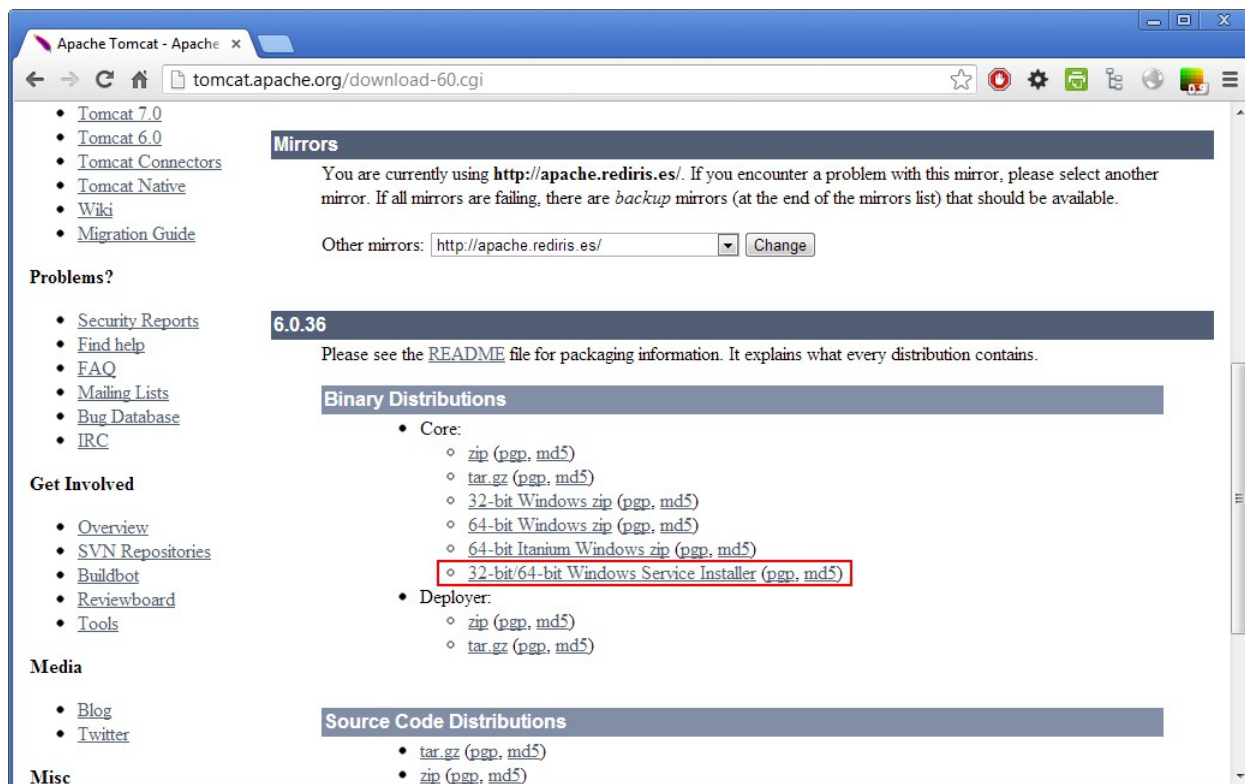


Figura 7.33. Pantalla y área de descarga del Contenedor Servlet Apache Tomcat

Una vez finalizada la descarga del archivo, hacemos doble clic sobre el mismo para iniciar el proceso de instalación del Contenedor Servlet.

La primera pantalla del asistente de instalación muestra una pantalla de bienvenida que informa que se va a iniciar el proceso de instalación (Figura 7.34) y en la que sólo tendremos que pulsar el botón **Next**.



Figura 7.34. Pantalla de bienvenida al proceso de instalación

La siguiente pantalla muestra el acuerdo de licencia (Figura 7.35). Pulsaremos en el botón **I Agree** para continuar con el proceso de instalación.

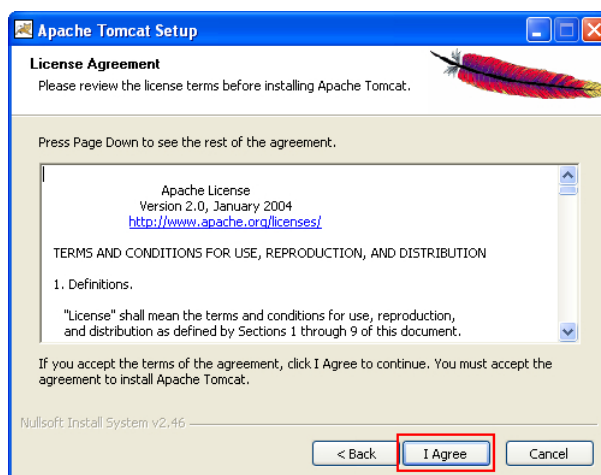


Figura 7.35. Pantalla de información del acuerdo de licencia

El siguiente paso del asistente nos mostrará los componentes de Apache Tomcat que queremos instalar (Figura 7.36). Podemos realizar la instalación normal o una personalizada si queremos instalar, por ejemplo, el **Host Manager** y/o los ejemplos incluidos o eliminar alguno de los componentes seleccionados por defecto. Una vez seleccionados los componentes a instalar pulsaremos el botón **Next**.

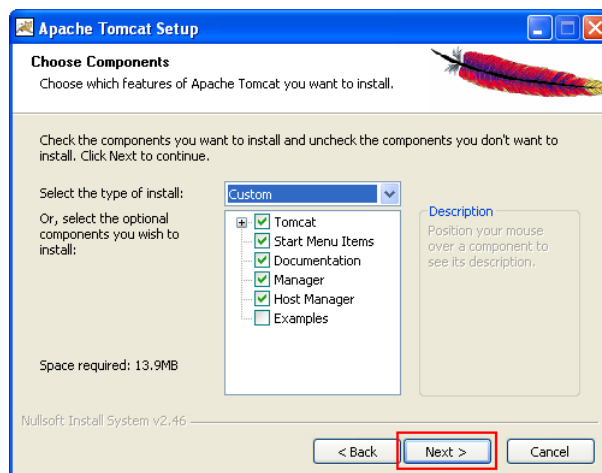


Figura 7.36. Pantalla de selección del tipo de instalación

A continuación se mostrará la pantalla donde configuraremos el servidor que podemos ver en la Figura 7.37. Para ello se nos pedirá introducir los puertos a utilizar por Tomcat, los cuales dejaremos con su valor por defecto, y el usuario administrador del servidor y los roles del mismo. Una vez introducidos todos estos datos volveremos a pulsar el botón **Next**.

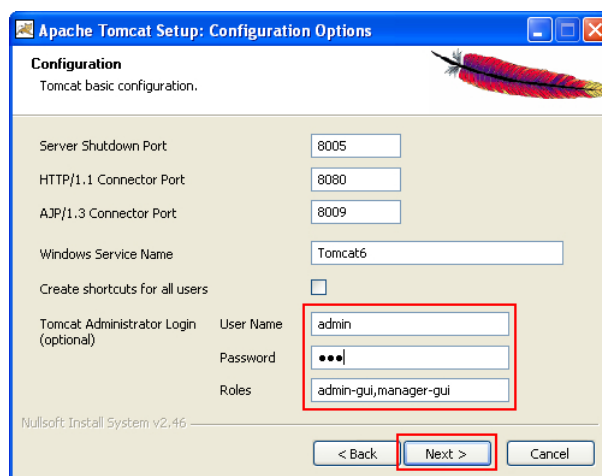


Figura 7.37. Pantalla de configuración del servidor y el usuario administrador

La siguiente pantalla nos pide indicar la ruta donde tenemos instalada la JRE a utilizar (Figura 7.38). Si tenemos instalada alguna debería detectarla por defecto, con lo que dejaremos dicho valor y pulsaremos el botón **Next**.

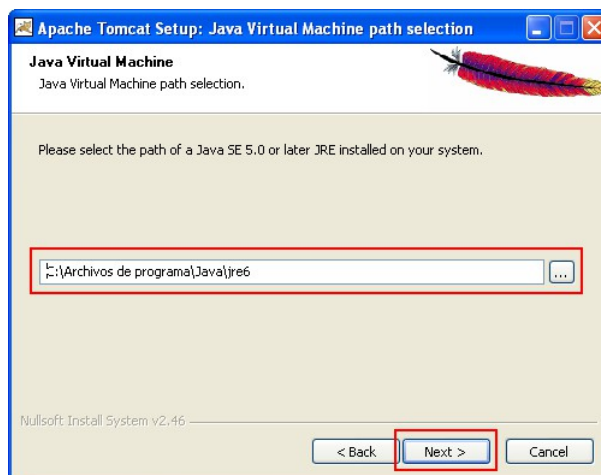


Figura 7.38. Pantalla de selección de la JRE a utilizar

Esta pantalla nos pide introducir o confirmar el directorio de instalación de Tomcat 6, tal como podemos ver en la figura 7.39. Dejaremos el valor por defecto y pulsaremos el botón **Install** para comenzar con la instalación del software.

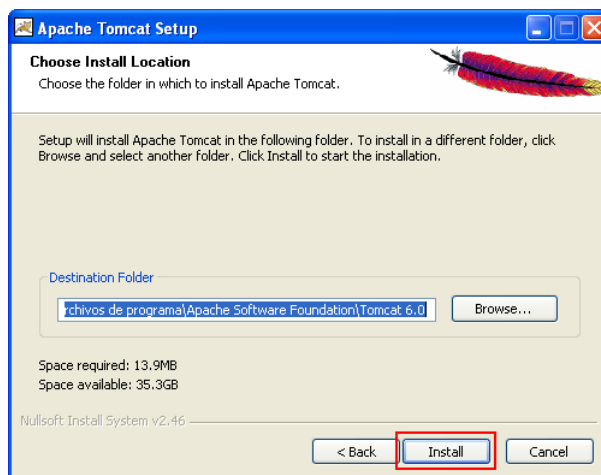


Figura 7.39. Pantalla de selección de la ruta de instalación

Una vez finalizada la instalación se mostrará la última pantalla del asistente, donde seleccionaremos **Run Apache Tomcat** para que se inicie el servicio y por tanto el servidor, y pulsaremos el botón **Finish**, tal como podemos ver en la Figura 7.40.

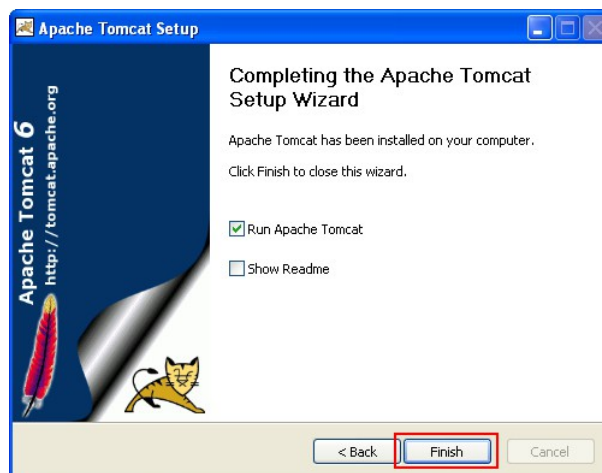


Figura 7.40. Pantalla final del proceso de instalación

Una vez arrancado el servidor, se mostrará en la barra de inicio de Windows un icono desde el cual podremos arrancar, parar o configurar nuestro servidor. Además, si accedemos en un navegador web a la dirección **http://localhost:8080/** podremos ver la página de inicio de Tomcat y validar que se ha instalado y arrancado correctamente.

El siguiente paso consiste en que, aunque la aplicación está preparada para trabajar sobre el Contenedor Servlet Tomcat 6, este no tiene instaladas las librerías de lenguaje de expresiones que permiten aprovechar todo el potencial proporcionado por JSF2 que si proporciona tomcat 7. Para permitir que la aplicación funcione sobre Apache Tomcat 6, tendremos que descargar las librerías **el-api-2.2.jar** y **el-imp-2.2.jar** de las URLs:

<http://www.java2s.com/Code/JarDownload/el/el-impl-2.2.jar.zip>

<http://www.java2s.com/Code/JarDownload/el/el-api-2.2.jar.zip>

y extraerlas y copiarlas en la carpeta **/lib** que se encuentra en el directorio de instalación de Apache Tomcat, en nuestro caso **C:\Archivos de programa\Apache Software Foundation\Tomcat 6.0\lib**.

El último paso a configurar dentro de nuestro Contenedor Servlet será instalar el conector Java para base de datos, ya que vamos a hacer que el pool de conexiones de la aplicación con MySQL sea gestionado por Apache Tomcat. Para ello detendremos el servidor de aplicaciones mediante el icono que se encuentra en la parte derecha de la barra de inicio de Windows.

Una vez detenido el servidor, copiaremos el fichero **mysql-connector-java-5.1.24-bin.jar** que se encuentra en el archivo ZIP que hemos descargado en el paso 7.2, en la carpeta **/lib** que se encuentra en el directorio de instalación de Apache Tomcat, en nuestro caso **C:\Archivos de programa\Apache Software Foundation\Tomcat 6.0\lib**. Una vez copiado el fichero podremos volver a arrancar el Contenedor.



7.5 Despliegue y configuración de la aplicación

El despliegue de la aplicación geec en el Contenedor Servlet lo realizaremos desde la consola de administración de Apache Tomcat. Para ello, abriremos un navegador web e introduciremos la URL **http://localhost:8080/manager/html**. Se mostrará el cuadro de diálogo que podemos ver en la Figura 7.41, donde debemos introducir el nombre de usuario y la contraseña del usuario administrador de Tomcat que hemos creado durante la instalación del mismo, y pulsar el botón **Aceptar**.



Figura 7.41. Diálogo de selección del usuario de acceso

Si hemos introducido los datos correctos, se mostrará la pantalla del **Gestor de aplicaciones web de Tomcat** que observamos en la Figura 7.42. Esta pantalla nos mostrará un listado de las aplicaciones desplegadas en el contenedor Servlet, y si nos desplazamos hacia la parte inferior de la misma, encontraremos el grupo de opciones de despliegue y en concreto la opción de **Archivo WAR a desplegar**.

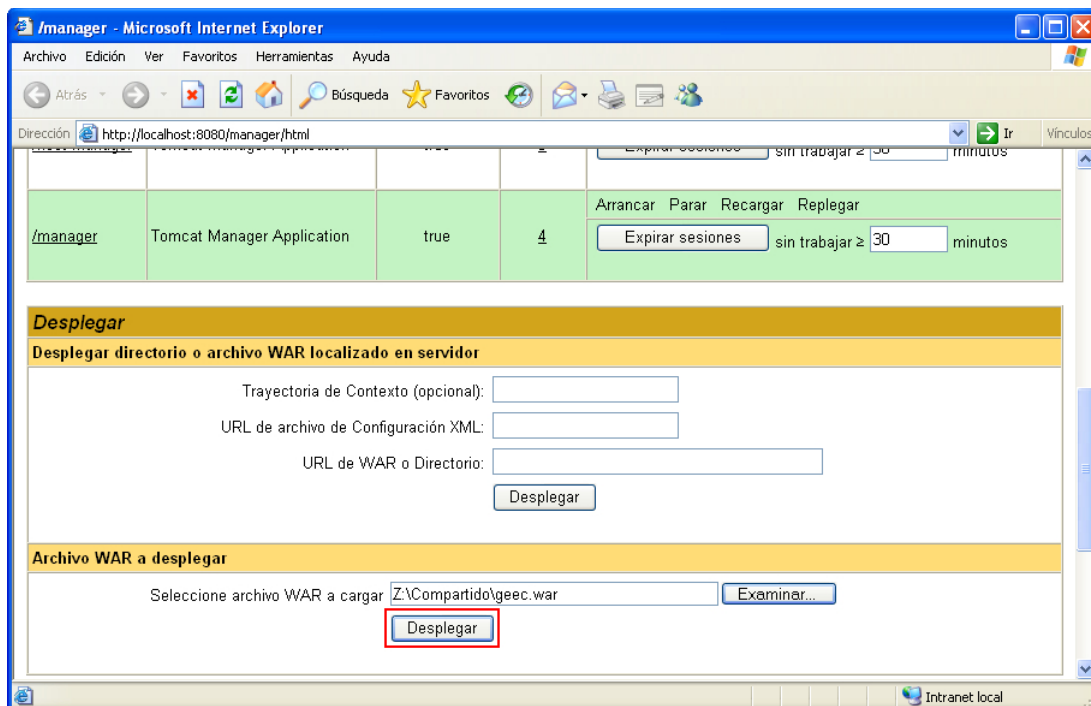


Figura 7.42. Pantalla inicial del Gestor de Aplicaciones Web de Tomcat

Una vez seleccionado el fichero **geec.war** pulsaremos el botón **Desplegar**. Una vez terminado el despliegue de la misma, podremos observar la sección **Aplicaciones**, donde debemos encontrar nuestra aplicación bajo el contexto web **/geec**, en estado detenido, tal como podemos ver en la figura 7.43.

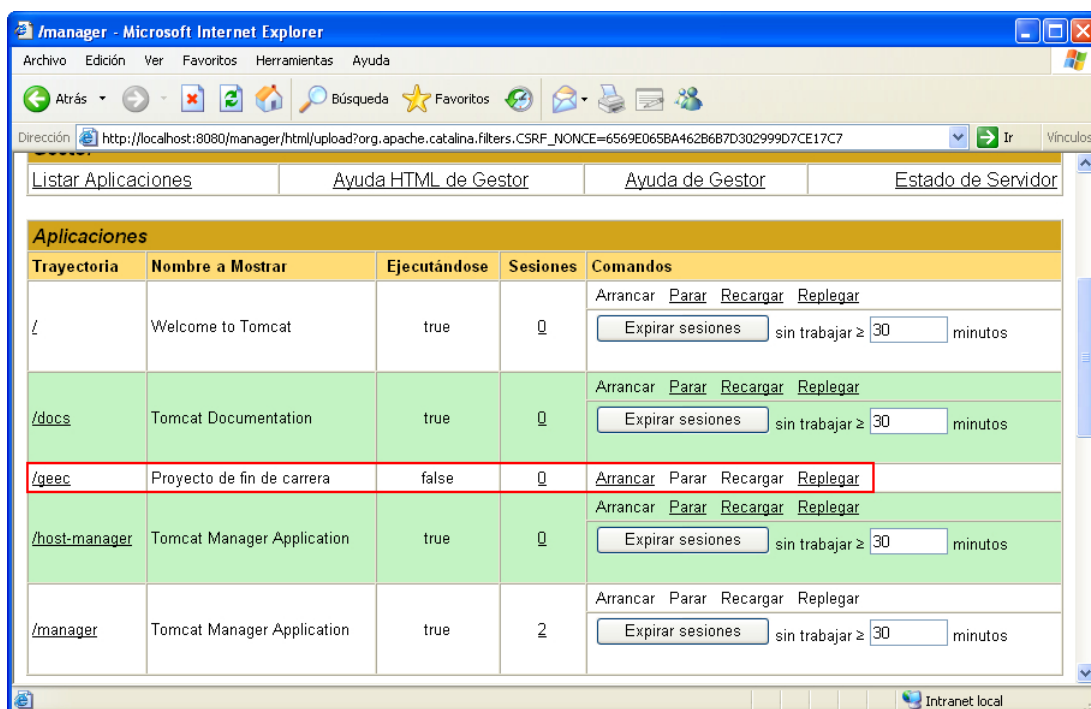


Figura 7.43. Resultado de la instalación y estado en el Gestor de aplicaciones web de Tomcat

Antes de arrancar la aplicación en el servidor debemos tener en cuenta una consideración con respecto a la configuración de la misma. Por defecto, la aplicación está configurada para conectarse a un SGBD ubicado en el mismo servidor (localhost), a un nombre de esquema denominado **geecdb** y con el usuario y contraseña **usuarioGeec** y **usuarioGeec** respectivamente. Si se ha configurado alguno de estos parámetros de forma distinta o se quiere cambiar la ubicación del servidor de base de datos por encontrarse en otra máquina habrá que modificar la configuración de la aplicación. Para ello, buscaremos el fichero **geec.xml** que se encuentra en la ruta **C:\Archivos de programa\Apache Software Foundation\Tomcat 6.0\conf\catalina\localhost**, tal como podemos observar en la figura 7.44.

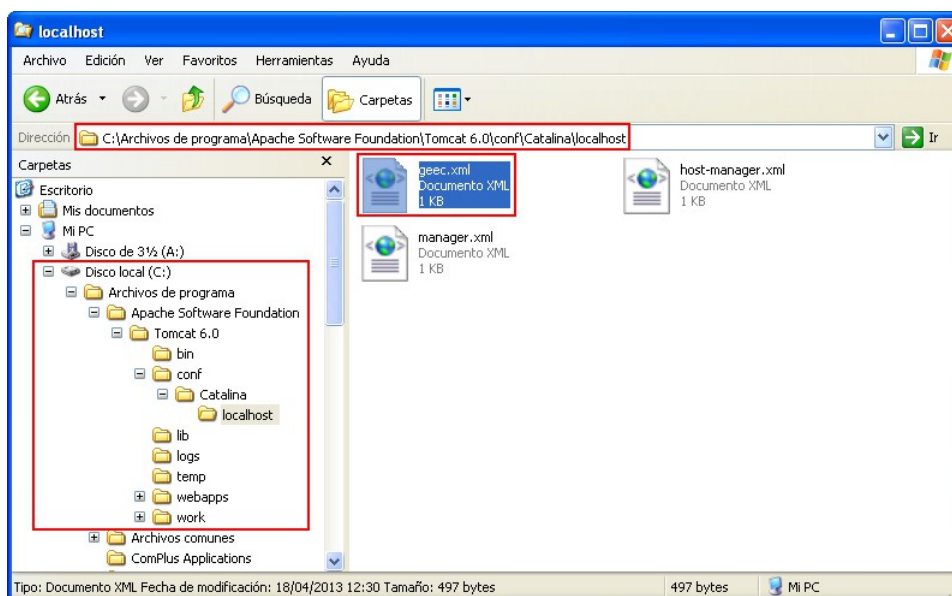


Figura 7.44. Ubicación de los ficheros de configuración de la conexión a BBDD de la aplicación

Editaremos el archivo con el programa wordpad u otro editor de texto de que dispongamos (Figura 7.45), modificaremos los parámetros que sean necesarios y guardaremos las modificaciones realizadas en el archivo de configuración.

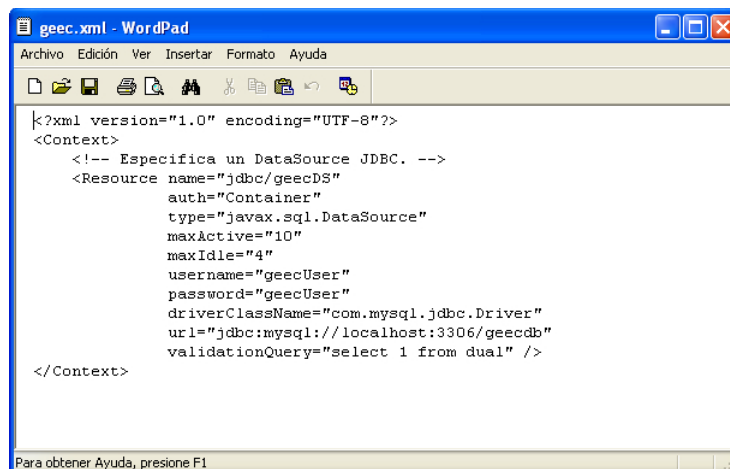


Figura 7.45. Ejemplo del contenido del fichero de configuración de la conexión a BBDD

Una vez desplegada la aplicación correctamente, y realizados los cambios necesarios, pulsaremos el enlace **Arrancar** correspondiente a nuestra aplicación del **Gestor de aplicaciones web de Tomcat** y ya tendremos iniciada la aplicación si todo funciona correctamente.

Para comprobarlo, abriremos otro navegador u otra pestaña del mismo e introduciremos la URL **<http://localhost:8080/geec>**, con lo que deberíamos ver la ventana de autenticación del sistema.



8 ANEXO C. Manual de usuario

El siguiente manual de usuario describe las distintas pantallas, agrupadas y/o agrupando funcionalidades, de que dispone la aplicación, explicando las distintas operaciones que se pueden realizar sobre cada una de ellas.

Durante el desarrollo del presente manual de uso de la aplicación se detallarán los usuarios que tienen acceso a cada una de las interfaces, teniendo en cuenta que el usuario Superadministrador podrá acceder a todas y cada una de las pantallas y funcionalidades de la aplicación. Además, como ejemplo durante todo este manual, se supondrá que se va a inicializar el sistema para gestionar los espacios y eventos culturales de los distintos Ayuntamientos de la Comunidad de Madrid.

8.1 Acceso al sistema

Para acceder al sistema de Gestión de Espacios y Eventos Culturales (en adelante GEEC), abriremos una ventana de un navegador (Microsoft IE, Google Chrome, Mozilla Firefox, Opera, etc...) e introduciremos en la barra de navegación la dirección de acceso proporcionada por el usuario administrador que se haya encargado de instalar la aplicación (será similar a `http://<dirección instalación>/geec`). Se mostrará el formulario de autenticación de usuario, que podemos ver en la Figura 8.1.

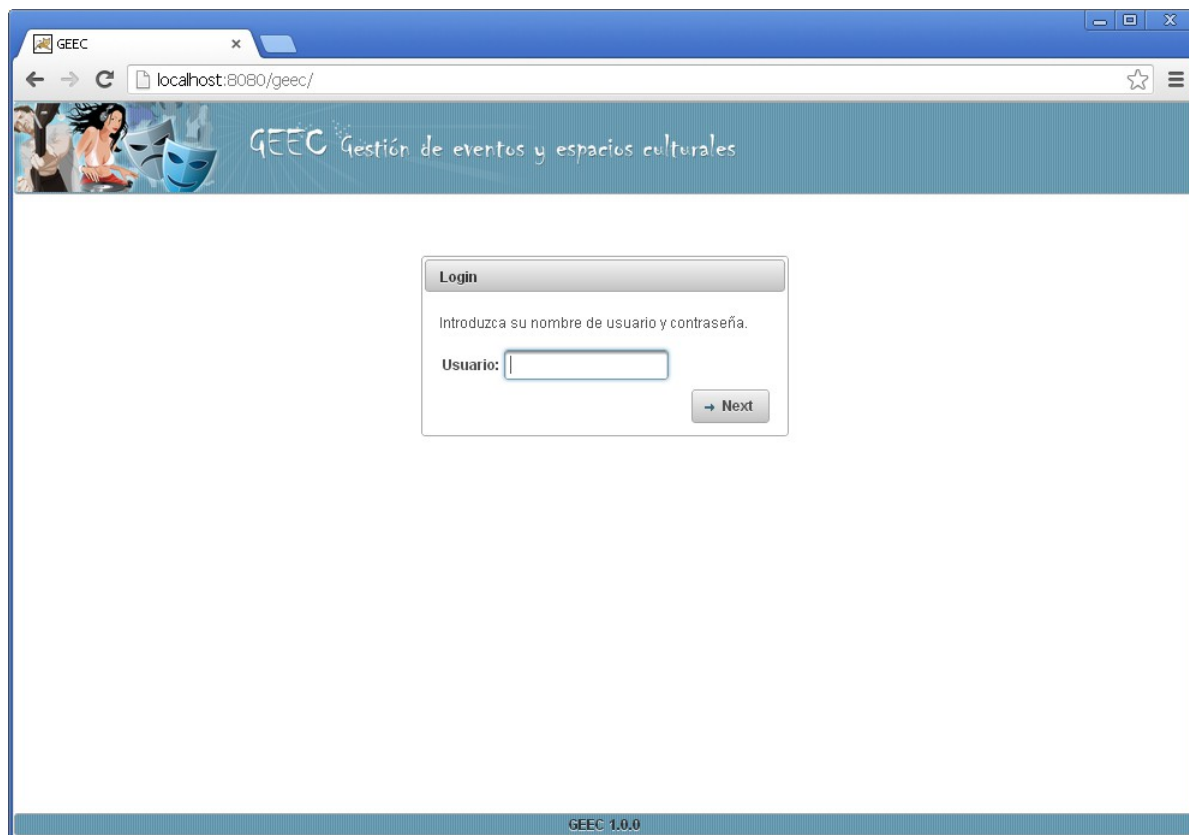


Figura 8.1. Pantalla de acceso a la aplicación. Introducción del usuario

En la primera pantalla del asistente de autenticación habrá que introducir el nombre de usuario y pulsar el botón **Next**. La segunda pantalla del asistente de autenticación solicitará la contraseña del usuario y la selección del grupo, en caso de que el usuario esté asociado a distintos grupos, con el cual interactuar en la aplicación y que podemos ver en la Figura 8.2. Una vez introducida la contraseña del usuario y el grupo a utilizar, habrá que pulsar el botón **Aceptar**.

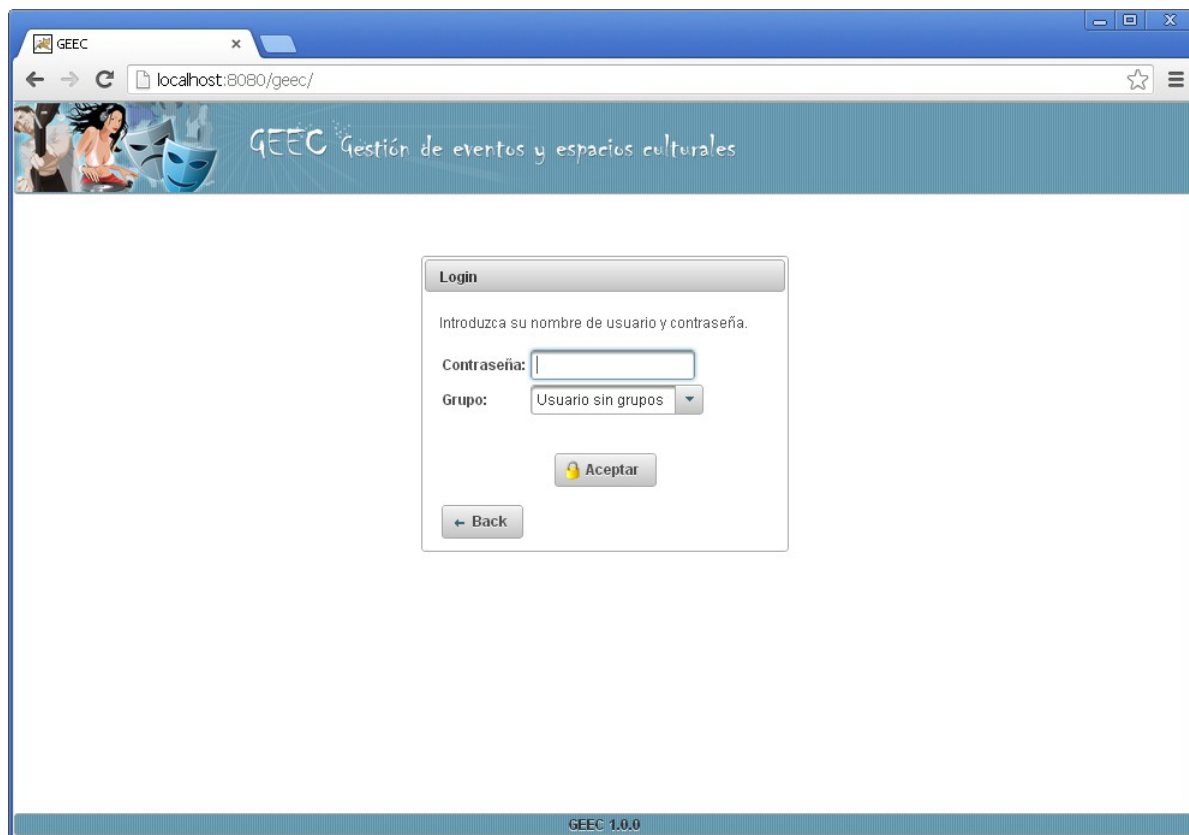


Figura 8.2. Pantalla de acceso a la aplicación. Introducción de contraseña y selección de grupo

Si el usuario y/o la contraseña son incorrectos se mostrarán los correspondientes mensajes de error. En caso contrario, se accederá a la aplicación con el perfil correspondiente al grupo seleccionado en la pantalla de autenticación, mostrando la pantalla principal de la misma que consistirá en la agenda de eventos para la semana en curso (Figura 8.3), aunque la primera vez que se acceda a la aplicación con el usuario administrador no aparecerá ninguna información en la agenda, ya que hay que inicializar el sistema con los datos de usuarios que pueden acceder a la misma para comenzar su explotación y que estos empiecen a introducir datos de espacios y eventos a gestionar.

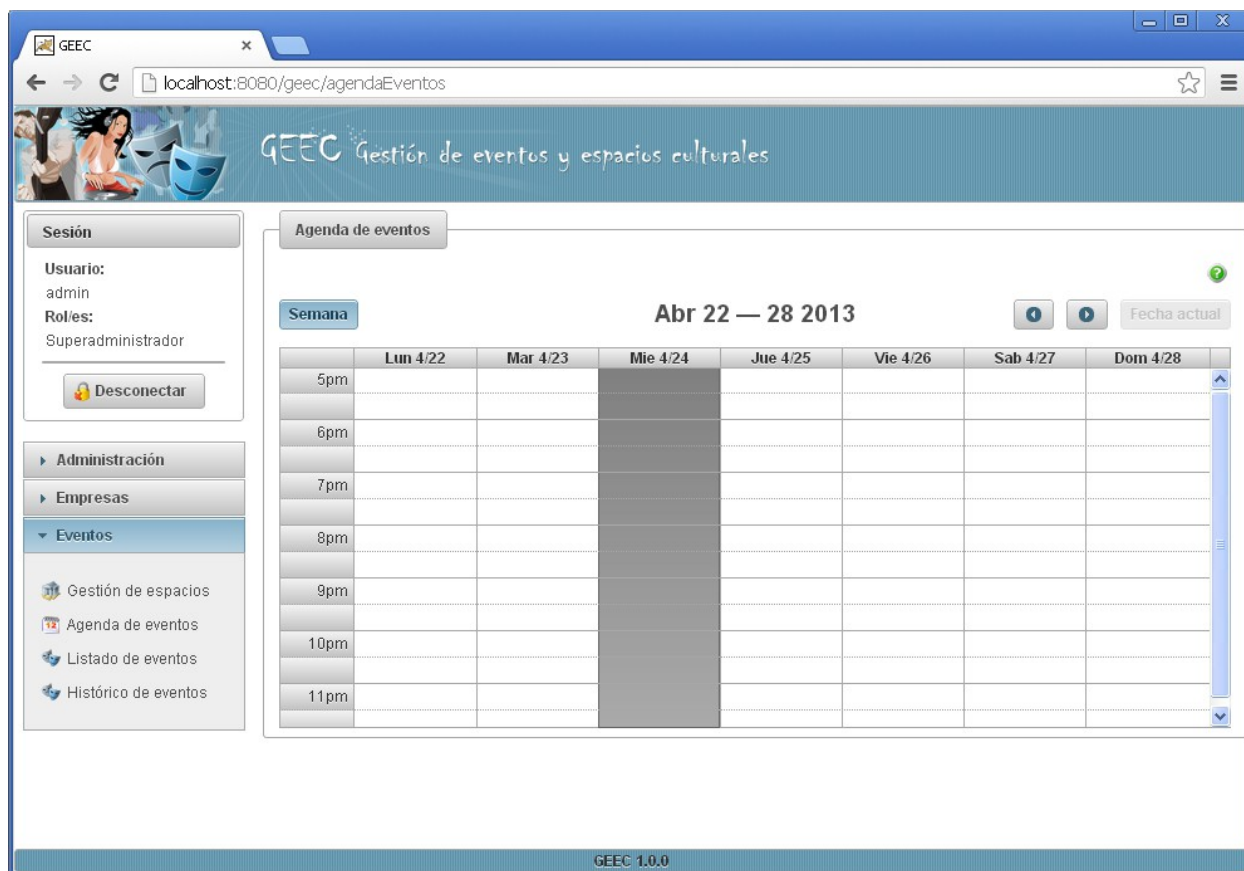


Figura 8.3. Pantalla principal de la aplicación. Agenda de eventos

El menú principal de la aplicación aparecerá en el lado izquierdo de la ventana y consistirá en dos partes diferenciadas. La parte superior será meramente informativa (Figura 8.4), con los datos del usuario autenticado en el sistema y los roles que desempeña en el grupo seleccionado.



Figura 8.4. Panel de información de sesión

La parte inferior del panel lateral izquierdo del sistema contendrá los distintos menús de operación en función del tipo usuario autenticado. Aunque en la Figura 8.5 se pueden observar que todos los menús mostrados y desplegados, sólo se mostrarán los menús relativos al usuario autenticado y en todo caso, sólo se mostrará desplegado el menú cuya opción se esté ejecutando actualmente (por defecto el menú de eventos).



Figura 8.5. Menú de operación

8.2 Administración del sistema

Este es el menú principal del administrador de sistema o Superadministrador. Aunque dicho usuario podrá prestar apoyo en cualquiera de las opciones del resto de tipos de usuario, su función principal es inicializar el sistema y gestionar los usuarios/empleados y grupos que podrán acceder a la aplicación.

8.2.1 Gestión de grupos

La primera opción de inicialización del sistema es la gestión de grupos de usuarios, a la cual sólo podrán acceder el o los usuarios con rol de Superadministrador del sistema. Para acceder a la misma habrá que seleccionar el menú **Administración** en el menú lateral y, dentro del mismo, seleccionar la opción **Gestión de grupos**. Esto mostrará la ventana de operaciones sobre los grupos de usuarios del sistema que podemos ver en la Figura 8.6.

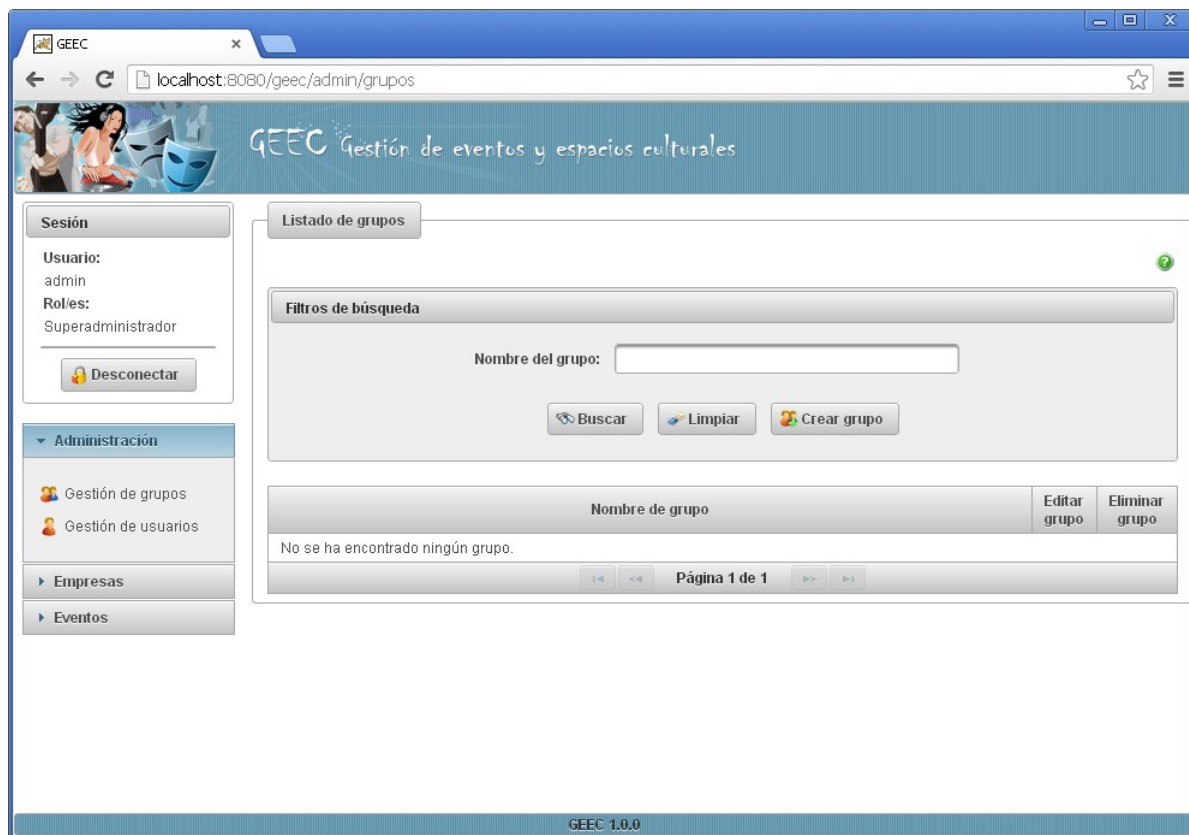


Figura 8.6. Pantalla de gestión de grupos

Sobre esta pantalla se pueden realizar las operaciones de búsqueda de grupos en función del nombre del mismo, búsqueda a la totalidad, creación de un nuevo grupo, edición de grupo y borrado de grupo. También se dispondrá de ayuda en pantalla para el usuario pulsando sobre el icono de ayuda que aparece en la esquina superior derecha de la pantalla. Si hacemos clic sobre dicho icono se mostrará el diálogo de información y ayuda de la pantalla, el cual estará disponible para el usuario en todo momento.

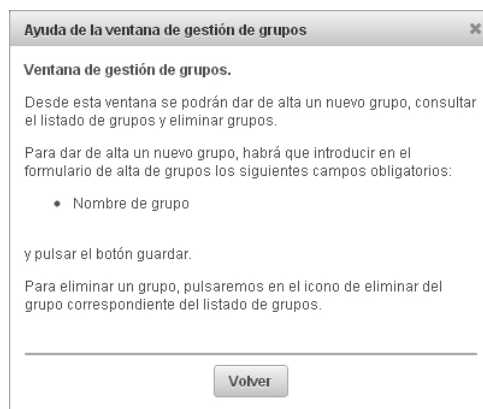


Figura 8.7. Diálogo de ayuda de la pantalla de gestión de grupos

8.2.1.1 Creación de un nuevo grupo de trabajo

El usuario Superadministrador podrá crear un nuevo grupo de trabajo pulsando sobre el botón **Crear grupo** (Figura 8.6), lo cual abrirá el cuadro de diálogo de creación de nuevo grupo (Figura 8.8).



El cuadro de diálogo 'Creación de grupo' tiene un título con un icono de cerrar. Contiene un campo de texto etiquetado 'Nombre del grupo: (*)' con un asterisco rojo indicando que es obligatorio. Debajo del campo hay dos botones: 'Guardar' y 'Cancelar'.

Figura 8.8. Cuadro de diálogo de creación de nuevo grupo

En este cuadro de diálogo habrá que introducir el nombre del grupo de trabajo y pulsar el botón **Guardar** para aceptar los datos introducidos y grabarlo en el sistema o **Cancelar** si queremos cancelar la creación. El campo nombre es obligatorio, y en caso de no introducirlo se mostrará el correspondiente mensaje de error. Una vez guardado el nuevo grupo de trabajo a crear, los cambios se reflejarán en el listado de grupos de la ventana principal de gestión de los mismos.

8.2.1.2 Edición de un grupo de trabajo

Si el sistema tiene registrado algún grupo de trabajo, en el listado de los mismos que se muestra en la ventana de gestión de grupos aparecerá, junto a cada uno de ellos, un icono de edición (Figura 8.9).

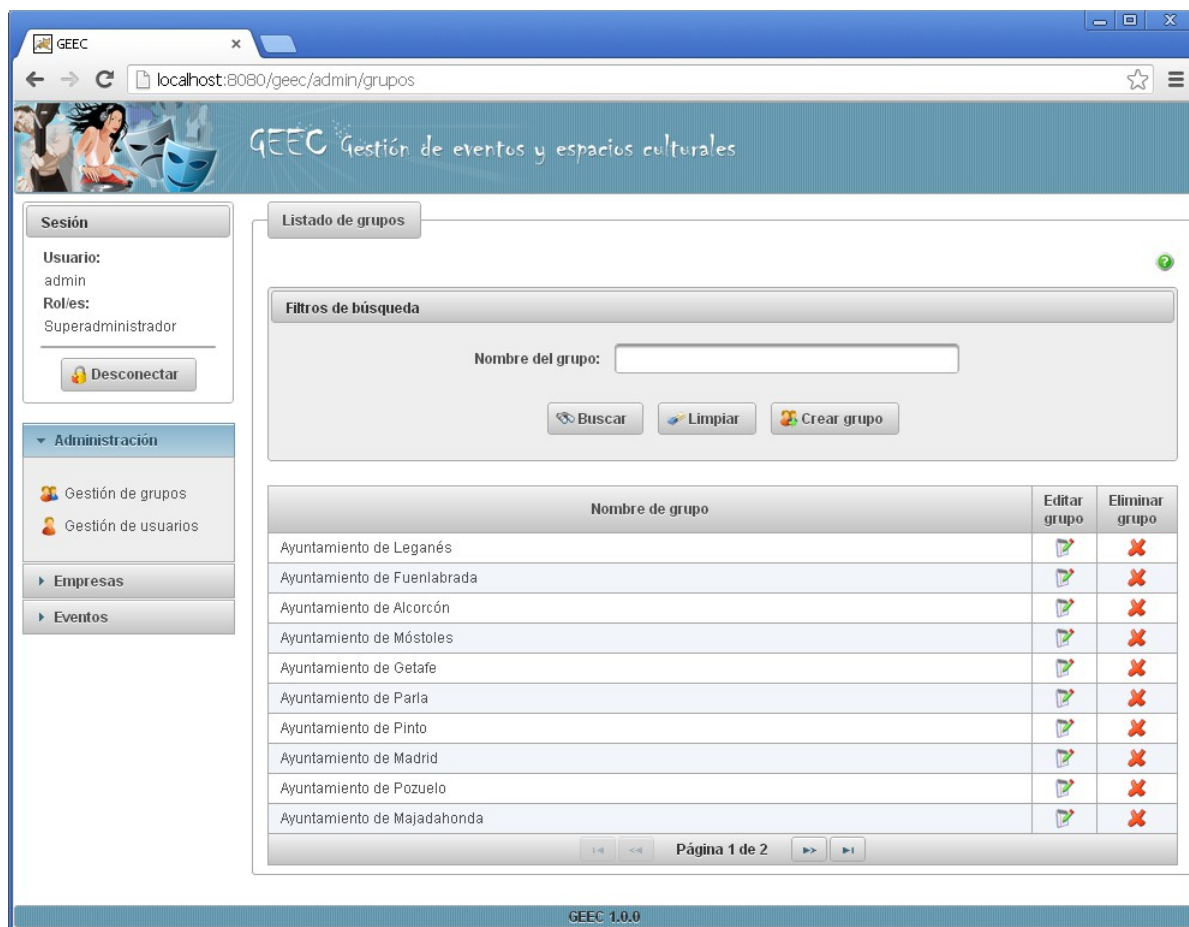


Figura 8.9. Pantalla de gestión de grupos de usuarios. Edición y borrado

Si pulsamos sobre el icono de edición de alguno de los grupos, se mostrará el diálogo de edición que podemos ver en la Figura 8.10.

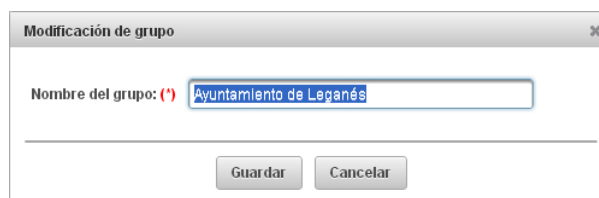


Figura 8.10. Diálogo de edición de grupo

En este formulario se podrá editar el nombre del grupo, el cual será obligatorio y no podrá ser igual que ningún otro nombre de grupo ya existente en la aplicación. En caso de dejar en blanco el nuevo nombre del grupo o de que sea igual que el nombre de un grupo ya existente se mostrará el correspondiente mensaje de error al pulsar el botón **Guardar**. En caso contrario, se actualizará el nombre del grupo y se

reflejarán los cambios en el elemento apropiado del listado de grupos.

8.2.1.3 Borrado de un grupo de trabajo

Si el sistema tiene registrado algún grupo de trabajo, en el listado de los mismos que se muestra en la ventana de gestión de grupos aparecerá, junto a cada uno de ellos, un icono de borrado (Figura 8.9). Si pulsamos sobre dicho icono, se mostrará el diálogo de confirmación de borrado del grupo de trabajo seleccionado (Figura 8.11).

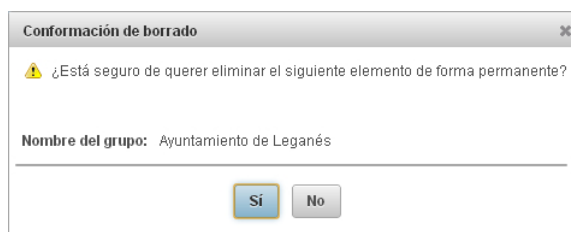


Figura 8.11. Diálogo de confirmación de borrado de grupo

Este cuadro de diálogo muestra un resumen de la información relativa al grupo de trabajo a eliminar, y proporciona al usuario dos botones de operación, el botón **Sí**, que eliminará el grupo, y el botón **No**, que cancelará la operación y volverá a la pantalla de listado de grupos.

Habrà que tener en cuenta que la operación de borrado no se podrá llevar a cabo en cualquiera de los siguientes supuestos:

- Si el grupo tiene usuarios asignados al mismo.
- Si el grupo tiene espacios (y/o eventos de espacios) asignados al mismo.

En cualquiera de estos supuestos se mostrarà el mensaje de error correspondiente.

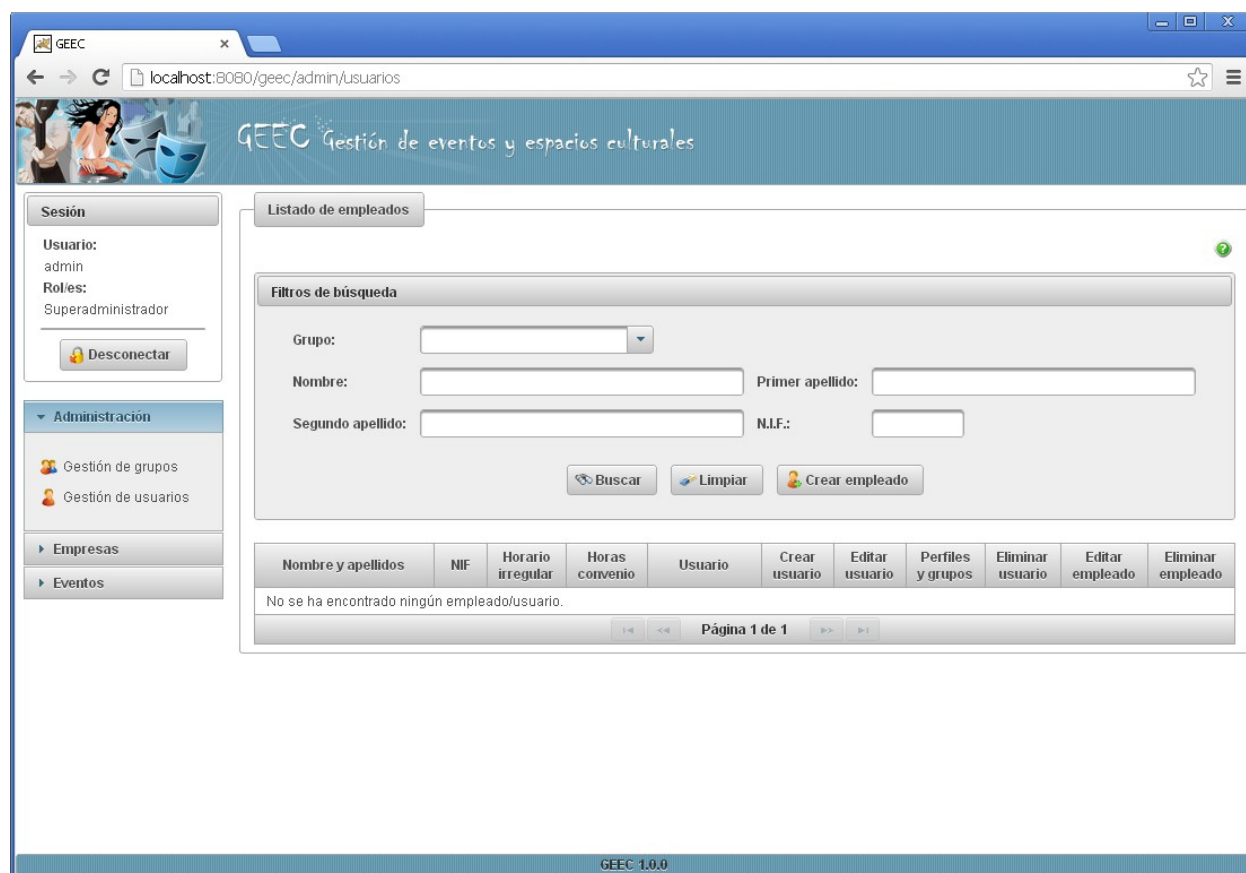
8.2.1.4 Búsqueda o filtro de grupos de trabajo

La página de gestión de grupos permite realizar búsquedas por el nombre completo o por partes del mismo. Para ello, introduciremos los parámetros de búsqueda en el campo nombre del formulario de búsqueda que aparece en la parte superior de la ventana (Figura 8.6) y pulsaremos el botón **Buscar**. Los resultados de la búsqueda aparecerán en el listado de la parte inferior de la ventana, si hay alguno, o en caso contrario se mostrarà el correspondiente mensaje informativo.

Para limpiar los parámetros de búsqueda pulsaremos el botón **Limpiar** y volverà a mostrarse el listado de grupos al completo.

8.2.2 Gestión de usuarios

La segunda opción de inicialización del sistema es la gestión de usuarios, a la cual sólo podrán acceder el o los usuarios con rol de Superadministrador del sistema o con rol Administrador de grupo. Para acceder a la misma habrá que seleccionar el menú **Administración** en el menú lateral y, dentro del mismo, seleccionar la opción **Gestión de usuarios**. Esto mostrará la ventana de operaciones sobre los usuarios del sistema que podemos ver en la Figura 8.12.



Nombre y apellidos	NIF	Horario irregular	Horas convenio	Usuario	Crear usuario	Editar usuario	Perfiles y grupos	Eliminar usuario	Editar empleado	Eliminar empleado
No se ha encontrado ningún empleado/usuario.										

Figura 8.12. Pantalla de gestión de usuarios

Sobre esta pantalla se pueden realizar las operaciones de búsqueda de empleados y usuarios en función del grupo a que pertenecen, del nombre y/o apellidos y del N.I.F. del mismo, búsqueda a la totalidad, creación de un nuevo empleado, edición de empleado, borrado de empleado, creación de usuario, edición de usuario, asignación de perfiles en grupos y eliminación de usuario. También se dispondrá de ayuda en pantalla para el usuario pulsando sobre el icono de ayuda que aparece en la esquina superior derecha de la pantalla. Si hacemos clic sobre dicho icono se mostrará el diálogo de información y ayuda de la pantalla, el cual estará disponible para el usuario en todo momento.

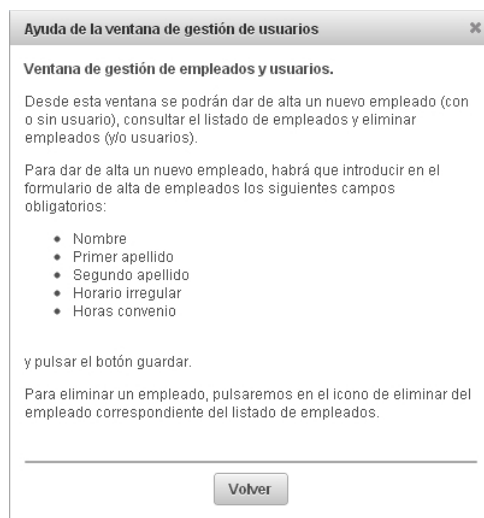


Figura 8.13. Diálogo de ayuda de la pantalla de gestión de usuarios

8.2.2.1 Creación de un nuevo empleado

El usuario Superadministrador podrá crear un nuevo empleado pulsando sobre el botón **Crear empleado** (Figura 8.12), lo cual abrirá el cuadro de diálogo de creación de nuevo empleado (Figura 8.14).

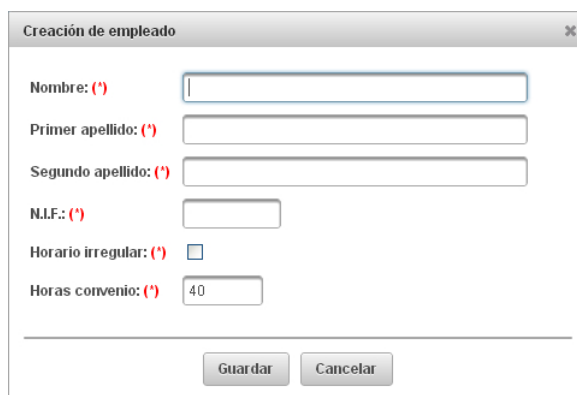
A screenshot of a "Creación de empleado" (Employee Creation) dialog box. It contains several input fields: "Nombre: (*)", "Primer apellido: (*)", "Segundo apellido: (*)", "N.I.F.: (*)", "Horario irregular: (*)" (with a checkbox), and "Horas convenio: (*)" (with a text box containing "40"). At the bottom, there are "Guardar" (Save) and "Cancelar" (Cancel) buttons.

Figura 8.14. Cuadro de diálogo de creación de nuevo empleado

En este cuadro de diálogo habrá que introducir el nombre y los apellidos del empleado, el N.I.F. E indicar si tiene un horario no estándar y las horas que tiene que cumplir por convenio, que por defecto serán 40h. Una vez rellenados estos datos pulsaremos el botón **Guardar** para aceptar los datos introducidos y grabarlo en el sistema o **Cancelar** si queremos cancelar la creación. En este cuadro de diálogo todos los campos son obligatorios, y en caso de no introducir alguno de ellos se mostrará el correspondiente mensaje de error. Una vez guardado el nuevo empleado a crear, los cambios se reflejarán en el listado de empleados de la ventana principal de gestión de los mismos.



8.2.2.2 Edición de un empleado

Si el sistema tiene registrado algún empleado, en el listado de los mismos que se muestra en la ventana de gestión de usuarios aparecerá, junto a cada uno de ellos, un icono de edición (Figura 8.15), únicamente para el usuario Superadministrador.

Nombre y apellidos	NIF	Horario irregular	Horas convenio	Usuario	Crear usuario	Editar usuario	Perfiles y grupos	Eliminar usuario	Editar empleado	Eliminar empleado
Jorge Vidal Alonso	69250648V	✗	40.00							
Juan Sastre Castillo	05824396Z	✗	40.00							
María Dolores Mariño Sánchez	01452604Q	✗	40.00							
Montserrat Martín Cervera	25607724F	✗	40.00							
Rafael Lorenzo Sánchez	68618100S	✗	40.00							

Figura 8.15. Pantalla de gestión de usuarios. Edición y borrado

Si pulsamos sobre el icono de edición de alguno de los empleados, se mostrará el diálogo de edición que podemos ver en la Figura 8.16.

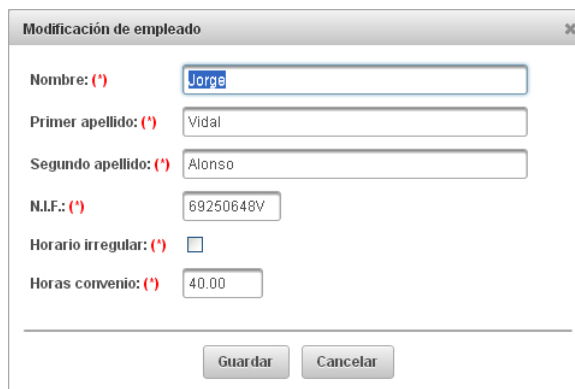


Figura 8.16. Diálogo de edición de empleado

En este formulario se podrá editar el nombre y los apellidos del empleado, el N.I.F. y los datos relativos al horario del empleado. En caso de dejar en blanco alguno de los campos o que el N.I.F. Introducido sea igual a uno ya existente se mostrarán los correspondientes mensajes de error al pulsar el botón **Guardar**. En caso contrario, se actualizará el empleado y se reflejarán los cambios en el elemento apropiado del listado de empleados y usuarios.

8.2.2.3 Borrado de un empleado

Si el sistema tiene registrado algún empleado, en el listado de los mismos que se muestra en la ventana de gestión de usuarios aparecerá, junto a cada uno de ellos, un icono de borrado (Figura 8.15). Si pulsamos sobre dicho icono, se mostrará el diálogo de confirmación de borrado del empleado seleccionado (Figura 8.17).

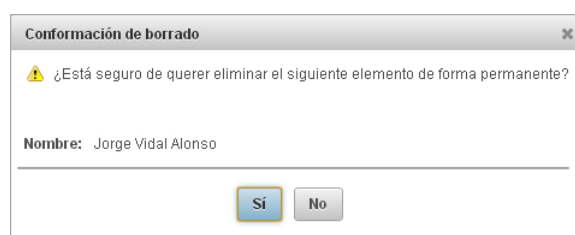


Figura 8.17. Diálogo de confirmación de borrado de empleado

Este cuadro de diálogo muestra un resumen de la información relativa al empleado a eliminar, y proporciona al usuario dos botones de operación, el botón **Sí**, que lo eliminará, y el botón **No**, que cancelará la operación y volverá a la pantalla de listado de empleados y usuarios.

Habrà que tener en cuenta que la operación de borrado implica los siguientes supuestos:

- Si se elimina un empleado que tiene asociado un usuario se eliminarà también el usuario asociado.



- Si se elimina un empleado con usuario que tiene roles en uno o varios grupos, se eliminarán todos sus roles en todos los grupos.

Una vez eliminado el empleado se mostrará el mensaje de información asociado y se reflejarán los cambios en el listado de empleados y usuarios.

8.2.2.4 Búsqueda o filtro de empleados

La página de gestión de usuarios permite realizar búsquedas por el nombre, primer apellido, segundo apellido, N.I.F. o por el grupo al cual pueda pertenecer el empleado. Para ello, introduciremos los parámetros de búsqueda en el campo apropiado del formulario de búsqueda que aparece en la parte superior de la ventana (Figura 8.15) y pulsaremos el botón **Buscar**. Los resultados de la búsqueda aparecerán en el listado de la parte inferior de la ventana, si hay alguno, o en caso contrario se mostrará el correspondiente mensaje informativo.

Para limpiar los parámetros de búsqueda pulsaremos el botón **Limpiar** y volverá a mostrarse el listado de empleados al completo.

8.2.2.5 Creación de usuario

El usuario Superadministrador podrá ver en el listado de empleados y usuarios la columna y el correspondiente icono de creación de usuario para cada uno de los empleados que todavía no tengan un usuario creado en la aplicación (Figura 8.15). Si seleccionamos dicho icono, se mostrará el diálogo de creación de nuevo usuario de empleado que se muestra en la Figura 8.18:

Creación de usuario

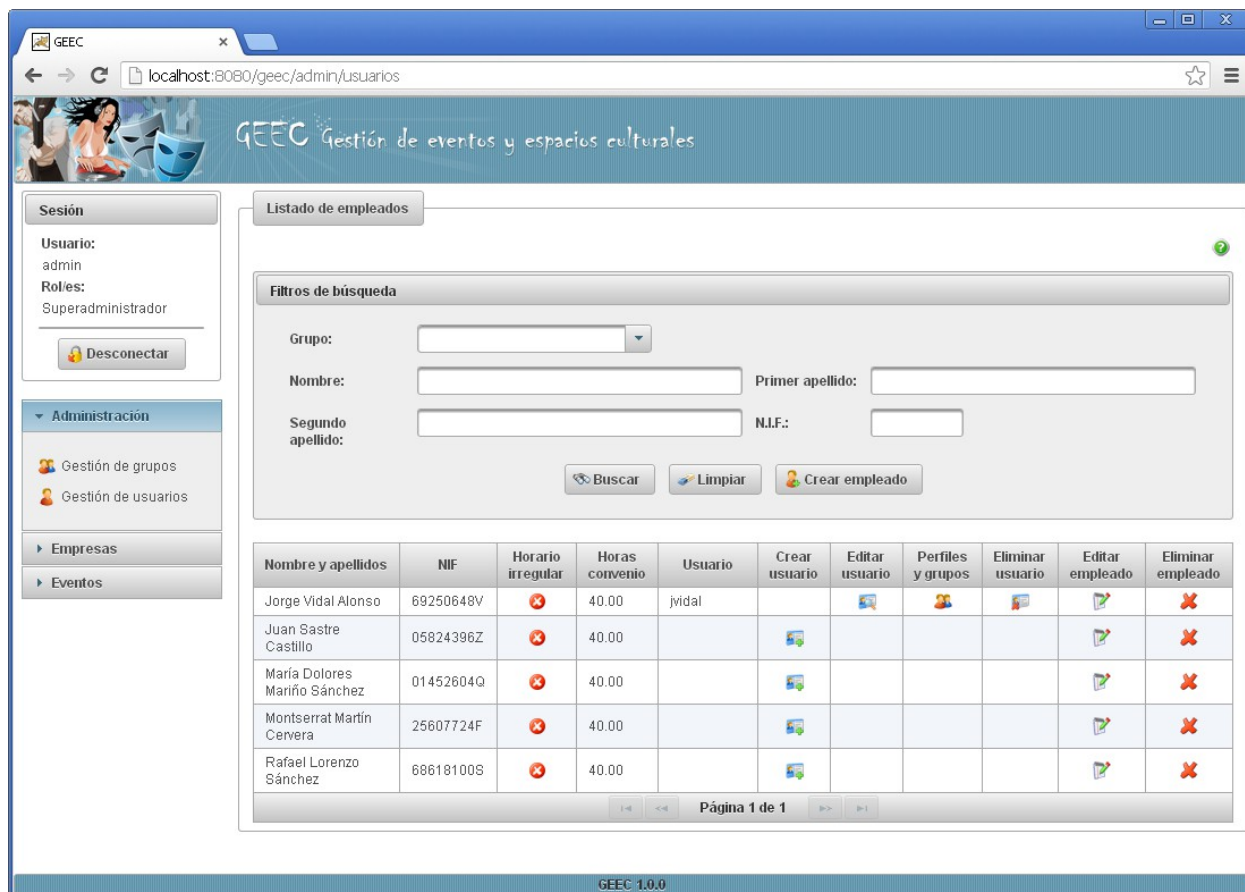
Nombre del usuario: (*)

Contraseña: (*)

Guardar Cancelar

Figura 8.18. Diálogo de creación de usuario

Habrà que introducir el nombre del usuario y la contraseña del mismo y pulsar el botón **Guardar** o **Cancelar** en función de si se quieren almacenar los cambios o no respectivamente. Una vez creado el usuario del empleado, se refrescarán los cambios en el listado, y junto al usuario que tenga usuario se mostrarán tres nuevos botones que permitirán modificar el usuario, eliminar el usuario y asignar roles en grupos para el mismo (Figura 8.19).



Sesión

Usuario: admin
Roles: Superadministrador
[Desconectar](#)

Administración

- Gestión de grupos
- Gestión de usuarios
- Empresas
- Eventos

Listado de empleados

Filtros de búsqueda

Grupo:
Nombre: Primer apellido:
Segundo apellido: N.I.F.:
[Buscar](#) [Limpiar](#) [Crear empleado](#)

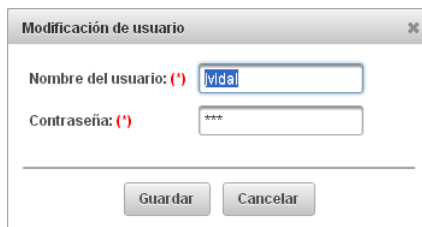
Nombre y apellidos	NIF	Horario irregular	Horas convenio	Usuario	Crear usuario	Editar usuario	Perfiles y grupos	Eliminar usuario	Editar empleado	Eliminar empleado
Jorge Vidal Alonso	69250648V		40.00	jvidal						
Juan Sastre Castillo	05824396Z		40.00							
María Dolores Mariño Sánchez	01452604Q		40.00							
Montserrat Martín Cervera	25607724F		40.00							
Rafael Lorenzo Sánchez	68618100S		40.00							

Página 1 de 1

Figura 8.19. Iconos de operación de usuario en el listado de empleados

8.2.2.6 Modificación de usuario

El usuario Superadministrador podrá modificar los datos de un usuario asignado a un empleado. Para ello pulsará en el botón de edición de usuario (Figura 8.19) y se mostrará el cuadro de diálogo de edición que podemos ver en la Figura 8.20:



Modificación de usuario

Nombre del usuario: (*)

Contraseña: (*)

[Guardar](#) [Cancelar](#)

Figura 8.20. Diálogo de edición de usuario

En este cuadro de diálogo habrá que introducir el nuevo nombre de usuario y/o la nueva contraseña y pulsar el botón **Guardar** si queremos almacenar los nuevos datos o el botón **Cancelar** si queremos descartar los cambios.



8.2.2.7 Borrado de usuario

El usuario Superadministrador podrá eliminar el usuario asignado a un empleado sin tener que eliminar todos los datos del empleado. Para ello, pulsará sobre el botón de eliminar usuario que se muestra en el listado de empleados, y se mostrará el diálogo de confirmación de borrado de usuario que podemos ver en la Figura 8.21:

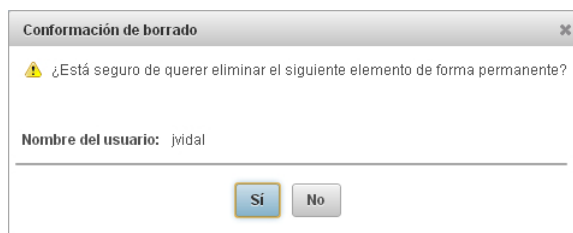


Figura 8.21. Diálogo de borrado de usuario

Pulsaremos en el botón **Sí** o **No**, en función de si queremos eliminar el usuario junto con los roles que tenga asignados en todos los grupos, o cancelar la operación.

8.2.3 Gestión de roles en grupos

Los usuarios Superadministrador y Administrador de grupo podrán gestionar los roles de los usuarios en un grupo de trabajo. Para ello, pulsarán el botón de gestionar roles de la columna Perfiles y Grupos del listado de empleados (Figura 8.19) para navegar a la ventana de gestión de perfiles y grupos del usuario seleccionado (Figura 8.22).

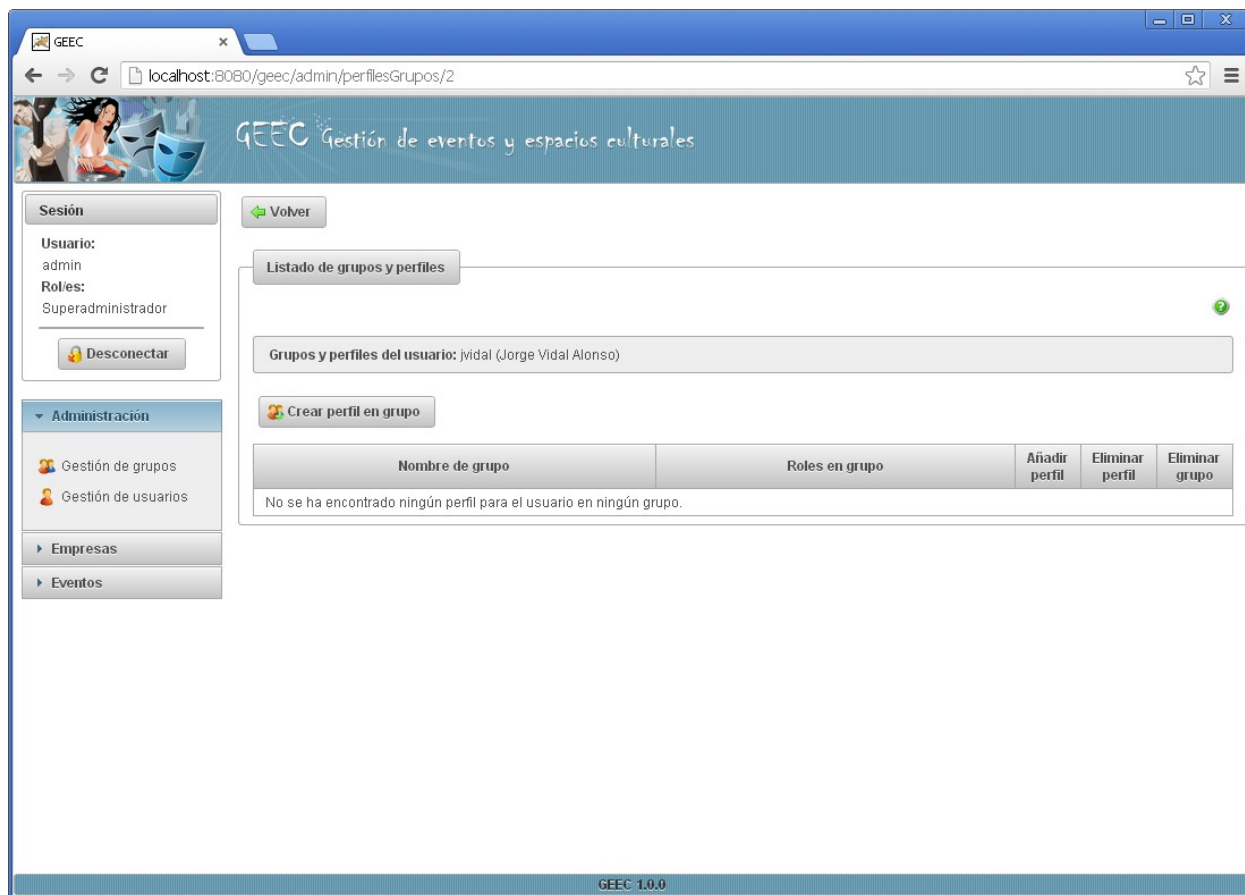


Figura 8.22. Ventana de gestión de perfiles y grupos de un usuario

En caso del usuario Superadministrador, éste podrá gestionar los distintos perfiles en todos los grupos disponibles en la aplicación, ya que tendrá acceso a todos ellos. En caso del usuario Administrador de grupo, sólo podrá gestionar los perfiles para el usuario en el grupo que administre.

Esta ventana también dispondrá del correspondiente diálogo de ayuda, al cual podremos acceder pulsando en el icono que se muestra en la esquina superior derecha de la misma (Figura 8.23), y que proporcionará al usuario ayuda contextual sobre la ventana en todo momento.

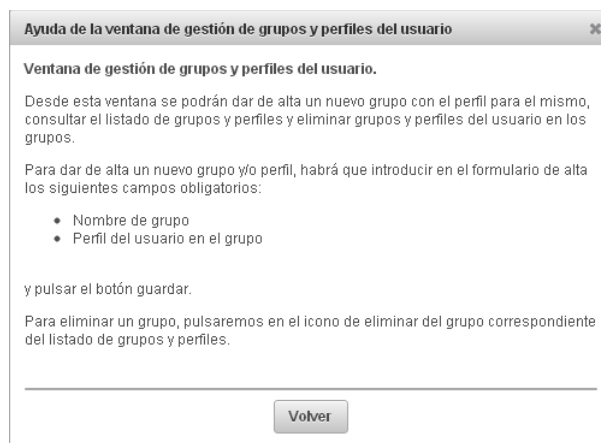


Figura 8.23. Diálogo de ayuda de la ventana de asignación de perfiles y grupos

8.2.3.1 Creación de perfil en grupo

El usuario Superadministrador podrá crear un nuevo perfil en un grupo para el usuario que se haya seleccionado en la pantalla de gestión de usuarios. Para ello pulsaremos en el botón de Crear perfil en grupo de la pantalla de gestión de perfiles y grupos (Figura 8.22), para abrir el diálogo de creación del nuevo perfil en grupo (Figura 8.24), y en él seleccionaremos el grupo para el cual asignar el perfil, y uno de los perfiles disponibles para el usuario en dicho grupo.



Figura 8.24. Diálogo de creación de perfil en grupo

Una vez seleccionado el grupo y el perfil, podremos guardar o cancelar los cambios con los botones **Guardar** y **Cancelar** respectivamente. Ambos campos son obligatorios, por lo que si no seleccionamos cualquiera de ellos se mostrarán los correspondientes mensajes de error. Una vez creado el nuevo perfil del usuario en el grupo seleccionado, se actualizará el listado de perfiles del usuario que aparece en la parte inferior de la pantalla, mostrando los nuevos iconos de operación para asignar nuevo perfil en el grupo, eliminar un perfil del grupo o eliminar todos los perfiles del grupo (Figura 8.25).

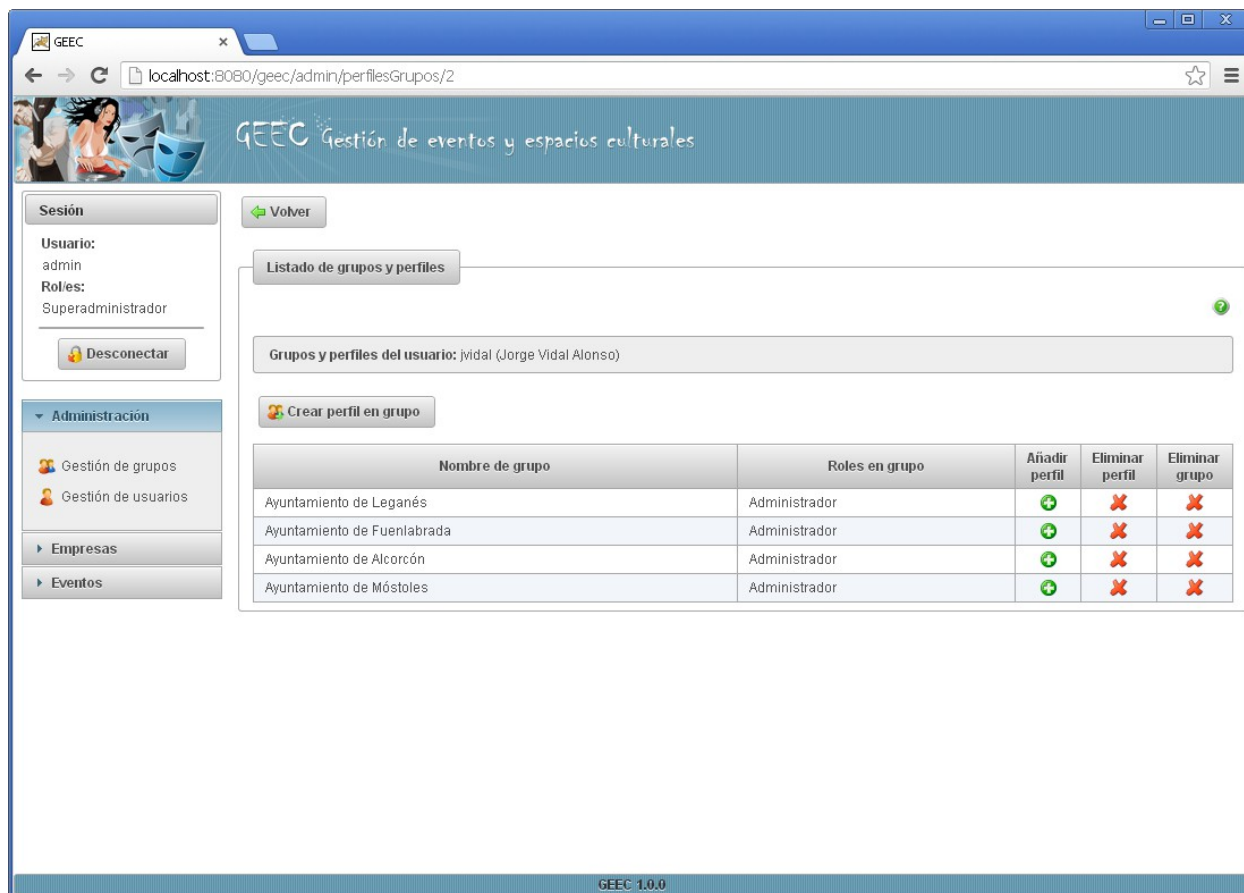


Figura 8.25. Listado de grupos y perfiles. Botones de operación por grupo

Hay que tener en cuenta que el usuario administrador no podrá gestionar sus propios perfiles, es decir, el usuario actualmente autenticado en la aplicación no verá los iconos de operación que afectan a su propio usuario.

8.2.3.2 Añadir perfil en grupo

El usuario Superadministrador y el Administrador de grupo podrán añadir un nuevo perfil para un usuario, en el caso del Superadministrador se podrá añadir el perfil para cualquier usuario en un grupo y en el caso del Administrador se podrá añadir el nuevo perfil para un usuario del grupo en el cual administra.

Para ello, pulsaremos el icono de la columna **Añadir perfil** correspondiente al grupo en el cual se quiere añadir el perfil (Figura 8.25), y se mostrará el diálogo de creación de perfil en grupo de la Figura 8.24. En este diálogo se seleccionará el nuevo perfil a añadir de la lista de perfiles y pulsaremos el botón **Guardar** para almacenar los cambios o **Cancelar** para rechazarlos. Los cambios se verán reflejados en el listado de grupos y perfiles del usuario.

8.2.3.3 Eliminar perfil de grupo

El usuario Superadministrador y el Administrador de grupo podrán eliminar un perfil de un usuario en un grupo.

Para ello, pulsaremos en el icono de eliminación de la columna **Eliminar perfil** del listado de grupos y perfiles del usuario seleccionado (Figura 8.25), y se mostrará el diálogo de eliminación de perfil en grupo que podemos ver en la Figura 8.26.

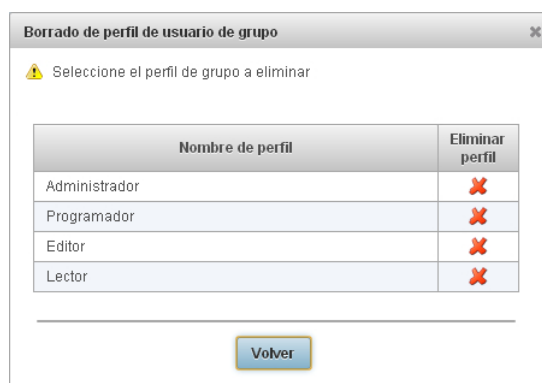


Figura 8.26. Diálogo de borrado de perfil en grupo

Sobre este diálogo podremos eliminar, de uno en uno, uno o varios perfiles del usuario en un determinado grupo pulsando sobre el icono de borrado. Una vez realizada la operación, pulsaremos el botón volver para regresar al listado de grupos y perfiles, donde podremos ver reflejados los cambios.

Hay que tener en cuenta que el usuario autenticado en la aplicación no podrá eliminar sus propios perfiles.

8.2.3.4 Eliminar grupo y perfiles

El usuario Superadministrador y el Administrador de grupo podrán eliminar todos los perfiles de un usuario en un grupo. Para ello, pulsarán el icono correspondiente de la columna **Eliminar grupo** del listado de perfiles y grupos (Figura 8.25). Esto mostrará el diálogo de confirmación de borrado del grupo al cual pertenece el usuario y todos los perfiles asociados al mismo que podemos ver en la Figura 8.27.

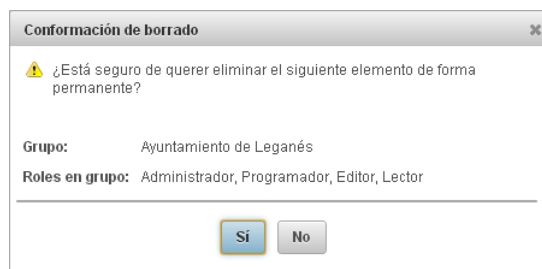


Figura 8.27. Diálogo de confirmación de borrado de grupo y perfiles

Si pulsamos el botón **Sí**, se eliminarán todos los perfiles que tenga el usuario en el grupo seleccionado. Si pulsamos **No** se cancelará la operación.

Hay que tener en cuenta que el usuario autenticado en la aplicación no puede eliminar sus propios perfiles en el grupo.

8.2.4 Gestión de empresas

Todos los usuarios de la aplicación podrán colaborar para mantener un registro de empresas y organizadoras de eventos y de contactos dentro de las mismas. Para acceder a la gestión de empresas habrá que seleccionar el submenú **Empresas** del menú lateral y, dentro del mismo, seleccionar la opción **Gestión de empresas**. Esto mostrará la ventana de operaciones sobre empresas que podemos ver en la Figura 8.28.

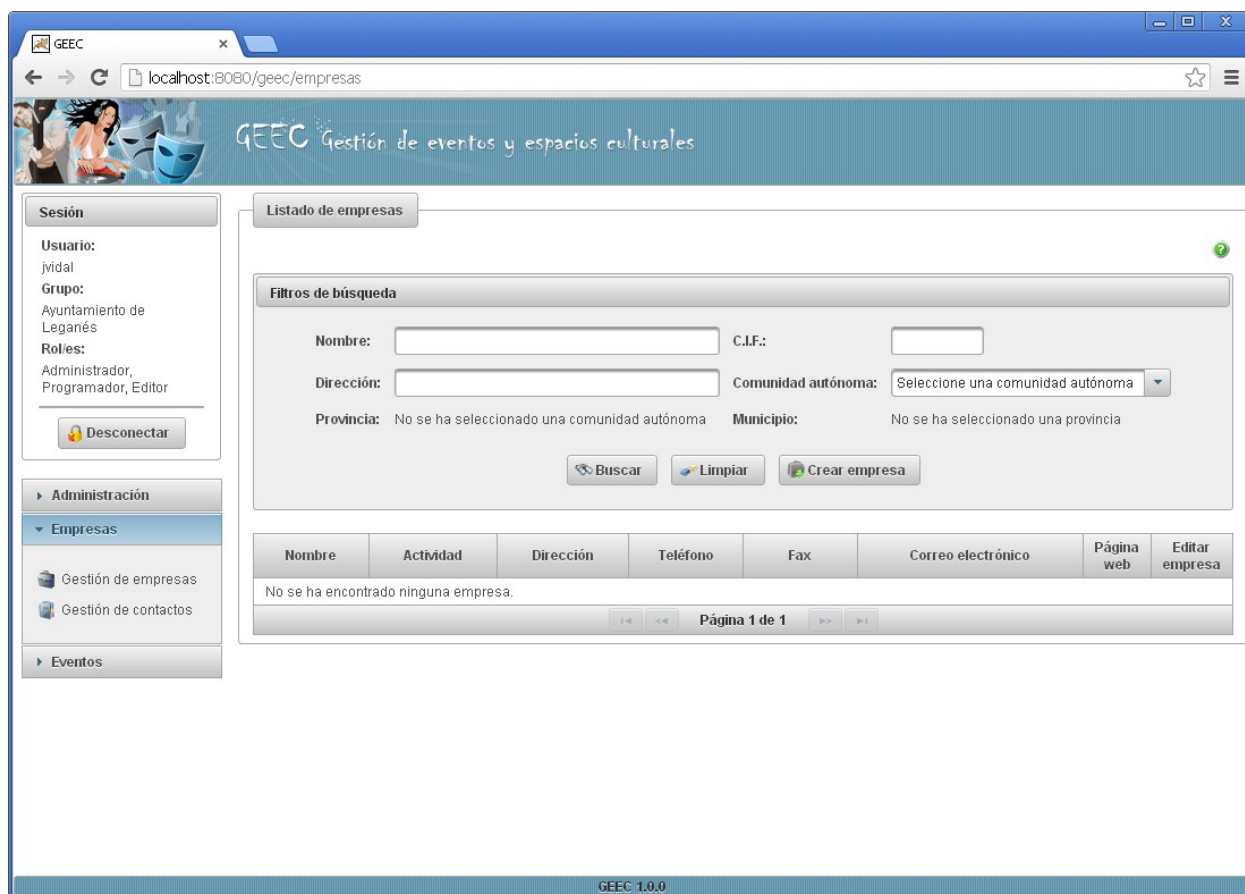


Figura 8.28. Pantalla de gestión de empresas organizadoras de eventos

Sobre esta pantalla se pueden realizar las operaciones de búsqueda de empresas en función del nombre de la misma, búsqueda a la totalidad, creación de una nueva empresa, edición de una empresa ya existente y borrado de empresas. También se dispondrá de ayuda en pantalla para el usuario pulsando

sobre el icono de ayuda que aparece en la esquina superior derecha de la pantalla. Si hacemos clic sobre dicho icono se mostrará el diálogo de información y ayuda de la pantalla, la cual estará disponible para el usuario en todo momento.

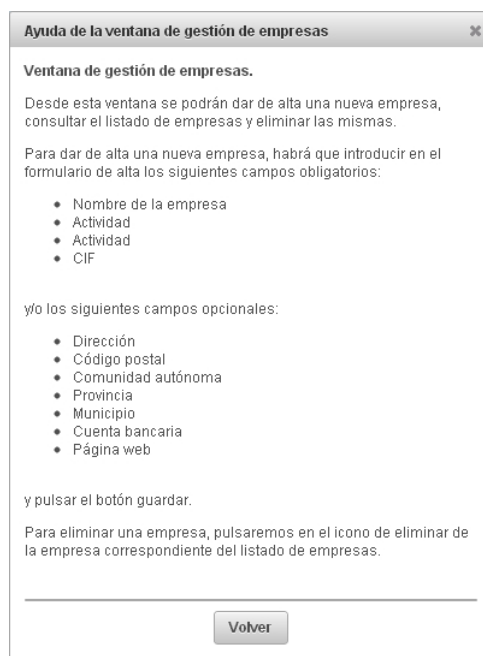


Figura 8.29. Diálogo de ayuda de la pantalla de gestión de empresas

8.2.4.1 Creación de una nueva empresa

Cualquier usuario podrá crear una nueva empresa pulsando sobre el botón **Crear empresa** (Figura 8.28), lo cual abrirá el cuadro de diálogo de creación de nueva empresa (Figura 8.30).

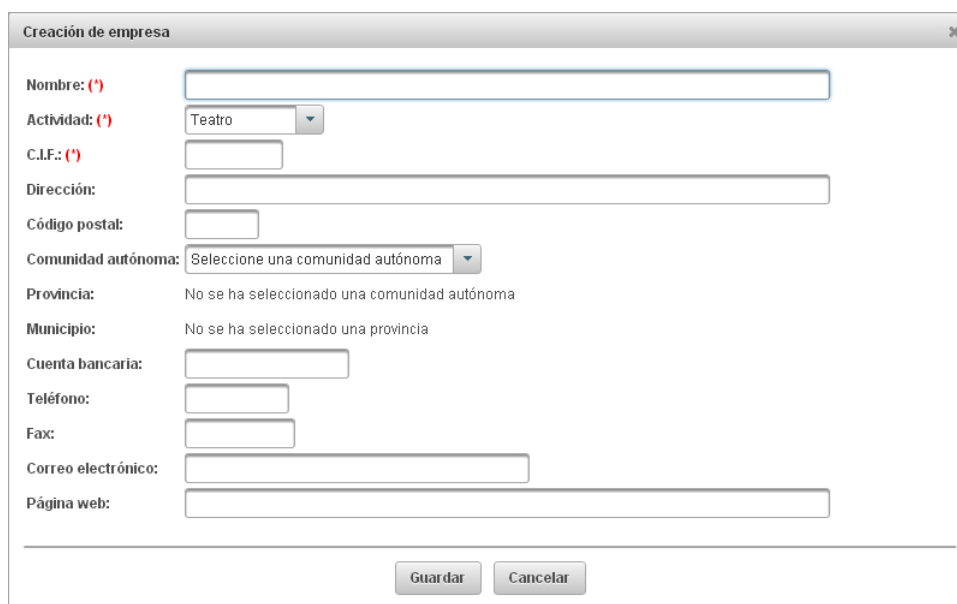


Figura 8.30. Diálogo de creación de nueva empresa



En este cuadro de diálogo habrá que introducir obligatoriamente el nombre de la empresa, la actividad que desempeña y el C.I.F. de la misma, el cual además deberá ser único, ya que en caso contrario se mostrarán los correspondientes mensajes de error. De forma opcional, para cada empresa se pueden almacenar los datos sobre su dirección, código postal, cuenta bancaria, teléfono y fax de contacto, correo electrónico y página web. Una vez introducidos los datos deseados, pulsaremos el botón **Guardar** y grabaremos en el sistema la nueva empresa o **Cancelar** si queremos cancelar la creación. Una vez guardada la nueva empresa, los cambios se reflejarán en el listado de empresas de la ventana principal.

8.2.4.2 Edición de empresa

Si el sistema tiene registrada alguna empresa de contacto, en el listado de empresas que se muestra en la ventana de gestión de empresas aparecerá, junto a cada una de ellas, un icono de edición (Figura 8.31).

Listado de empresas

Filtros de búsqueda

Nombre: C.I.F.:

Dirección: Comunidad autónoma:

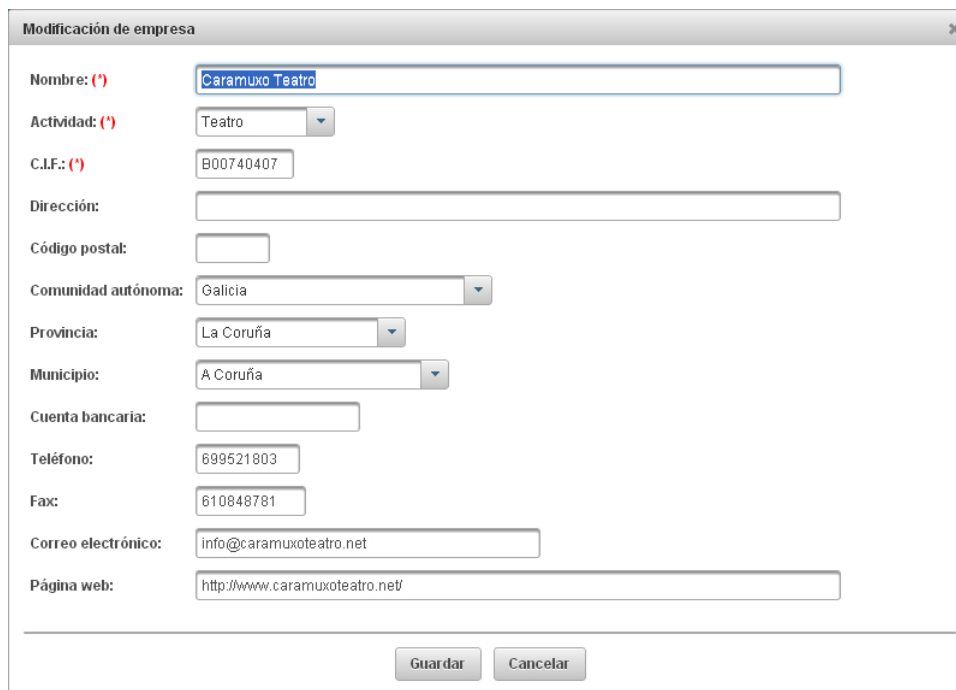
Provincia: No se ha seleccionado una comunidad autónoma Municipio: No se ha seleccionado una provincia

Nombre	Actividad	Dirección	Teléfono	Fax	Correo electrónico	Página web	Editar empresa	Eliminar empresa
Caramuxo Teatro	Teatro	A Coruña (La Coruña)	699521803	610848781	info@caramuxoteatro.net			
Orden y concierto S.L.	Música	Suero de Quiñones, 4, 1º C 28002 - Madrid (Madrid)	686477440		info@ordenyconcierto.com			
Producciones Yllana	Teatro	C/ Madera, 29 28004 - Madrid (Madrid)	915233301	915233309	infoyllana@yllana.com			
Rol Lalá	Teatro				ronlala@ronlala.com			

Página 1 de 1

Figura 8.31. Pantalla de gestión de empresas. Edición y borrado

Si pulsamos sobre el icono de edición de alguna de las empresas, se mostrará el diálogo de edición que podemos ver en la Figura 8.32.



Modificación de empresa

Nombre: (*) Caramuxo Teatro

Actividad: (*) Teatro

C.I.F.: (*) B00740407

Dirección:

Código postal:

Comunidad autónoma: Galicia

Provincia: La Coruña

Municipio: A Coruña

Cuenta bancaria:

Teléfono: 699521803

Fax: 610848781

Correo electrónico: info@caramuxoteatro.net

Página web: http://www.caramuxoteatro.net/

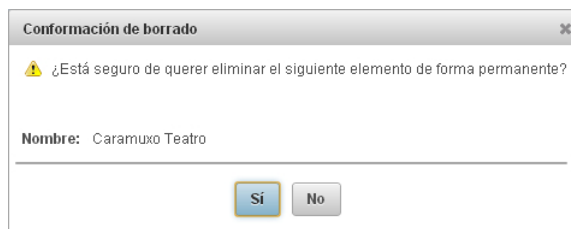
Guardar Cancelar

Figura 8.32. Diálogo de edición de empresa

En este formulario se podrán editar todos los campos de la empresa. En caso de dejar en blanco alguno de los campos obligatorios o de que el C.I.F. ya exista se mostrará el correspondiente mensaje de error al pulsar el botón **Guardar**. En caso contrario, se actualizará la empresa y se reflejarán los cambios en el elemento apropiado del listado de empresas.

8.2.4.3 Borrado de empresa

Si el sistema tiene registrada alguna empresa de contacto, en el listado de las mismas que se muestra en la ventana de gestión de empresas aparecerá, junto a cada una de ellas, un icono de borrado (Figura 8.31). Si pulsamos sobre dicho icono, se mostrará el diálogo de confirmación de borrado de empresa la seleccionada (Figura 8.33).



Confirmación de borrado

⚠ ¿Está seguro de querer eliminar el siguiente elemento de forma permanente?

Nombre: Caramuxo Teatro

Sí No

Figura 8.33. Diálogo de confirmación de borrado de empresa

Este cuadro de diálogo muestra un resumen de la información relativa a la empresa a eliminar, y



proporciona al usuario dos botones de operación, el botón **Si**, que eliminará la empresa, y el botón **No**, que cancelará la operación y volverá a la pantalla de listado de empresas de contacto.

Habrà que tener en cuenta que la operación de borrado no se podrá llevar a cabo en cualquiera de los siguientes supuestos:

- Si la empresa está asignada a algún contacto.
- Si la empresa tiene eventos asignados a la misma.

En cualquiera de estos supuestos se mostrarà el mensaje de error correspondiente.

8.2.4.4 *Búsqueda o filtro de empresas*

La página de gestión de empresas permite realizar búsquedas por el nombre, C.I.F., dirección (dirección postal, comunidad, provincia, municipio). Para ello, introduciremos los parámetros de búsqueda en el campo correspondiente del formulario de búsqueda que aparece en la parte superior de la ventana (Figura 8.31) y pulsaremos el botón **Buscar**. Los resultados de la búsqueda aparecerán en el listado de la parte inferior de la ventana, si hay alguno, o en caso contrario se mostrarà el correspondiente mensaje informativo.

Para limpiar los parámetros de búsqueda pulsaremos el botón **Limpiar** y volverà a mostrarse el listado de empresas al completo.

8.2.5 *Gestión de contactos*

Para acceder a la gestión de contactos de empresa habrá que seleccionar el submenú **Empresas** del menú lateral y, dentro del mismo, seleccionar la opción **Gestión de contactos**. Esto mostrarà la ventana de operaciones sobre contactos que podemos ver en la Figura 8.34.

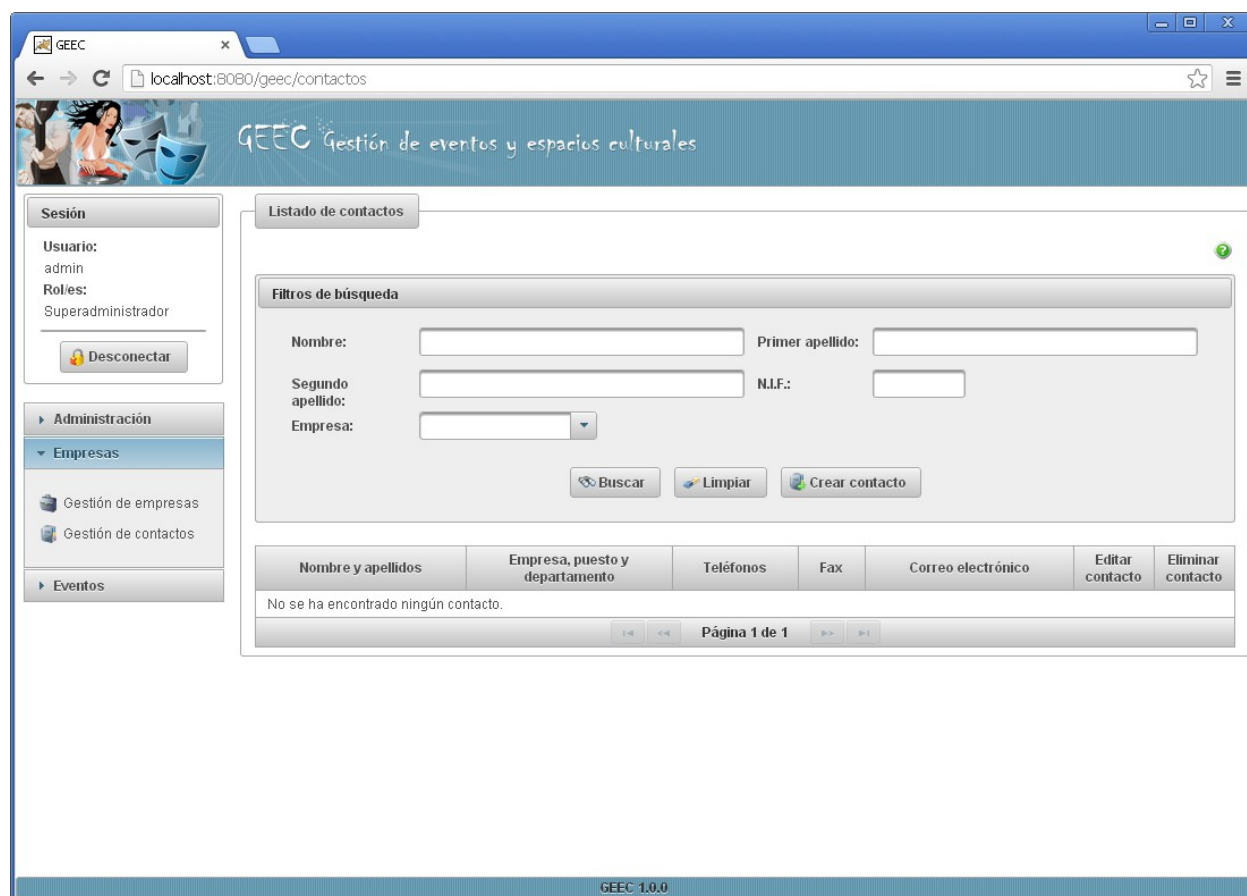


Figura 8.34. Pantalla de gestión de contactos de empresa

Sobre esta pantalla se pueden realizar las operaciones de búsqueda de contactos con filtros, búsqueda a la totalidad, creación de un nuevo contacto, edición de un contacto ya existente y borrado de contactos. También se dispondrá de ayuda en pantalla para el usuario pulsando sobre el icono de ayuda que aparece en la esquina superior derecha de la pantalla. Si hacemos clic sobre dicho icono se mostrará el diálogo de información y ayuda de la pantalla, la cual estará disponible para el usuario en todo momento.

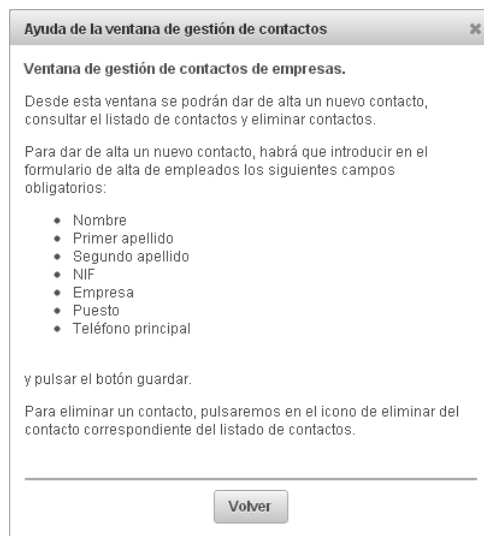
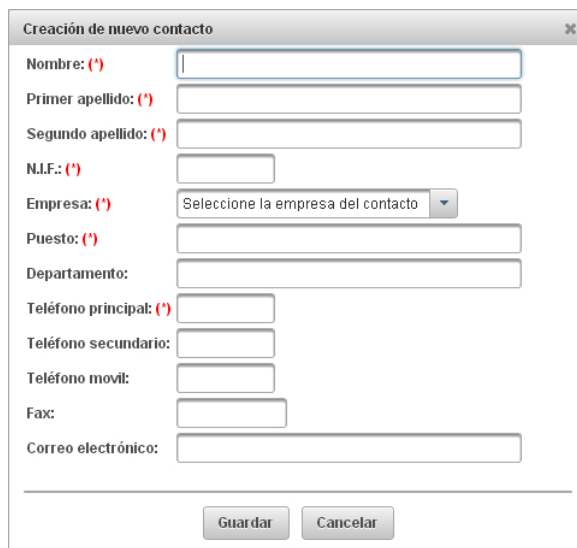


Figura 8.35. Diálogo de ayuda de la pantalla de gestión de contactos

8.2.5.1 Creación de un nuevo contacto

Cualquier usuario podrá crear un nuevo contacto pulsando sobre el botón **Crear contacto** (Figura 8.34), lo cual abrirá el cuadro de diálogo de creación de contacto que podemos ver en la siguiente figura:



Creación de nuevo contacto

Nombre: (*)

Primer apellido: (*)

Segundo apellido: (*)

N.I.F.: (*)

Empresa: (*) Seleccione la empresa del contacto

Puesto: (*)

Departamento:

Teléfono principal: (*)

Teléfono secundario:

Teléfono móvil:

Fax:

Correo electrónico:

Guardar Cancelar

Figura 8.36. Diálogo de creación de nuevo contacto

En este cuadro de diálogo habrá que introducir obligatoriamente el nombre y los apellidos del contacto, el N.I.F., el cual además deberá ser único ya que en caso contrario se mostrarán los correspondientes mensajes de error, la empresa a la que pertenece y el puesto que ocupa y el teléfono de contacto. De forma opcional, para cada contacto se pueden almacenar los datos sobre su departamento, teléfonos alternativos, fax y correo electrónico. Una vez introducidos los datos deseados, pulsaremos el botón



Guardar y grabaremos en el sistema el nuevo contacto o **Cancelar** si queremos cancelar la creación. Una vez guardado el contacto, los cambios se reflejarán en el listado de contactos de la ventana principal.

8.2.5.2 Edición de un contacto

Si el sistema tiene registrado algún contacto de empresa, en el listado de contactos que se muestra en la ventana de gestión de contactos aparecerá, junto a cada uno de ellos, un icono de edición (Figura 8.37).

The screenshot shows the GEEC web application interface. The browser address bar displays 'localhost:8080/geec/contactos'. The page title is 'GEEC Gestión de eventos y espacios culturales'. The sidebar on the left contains the following sections:

- Sesión**: Usuario: admin, Roles: Superadministrador, Desconectar button.
- Administración**
- Empresas**: Gestión de empresas, Gestión de contactos.
- Eventos**

The main content area is titled 'Listado de contactos'. It features a 'Filtros de búsqueda' section with the following fields:

- Nombre:
- Primer apellido:
- Segundo apellido:
- N.I.F.:
- Empresa:

Below the filters are three buttons: 'Buscar', 'Limpiar', and 'Crear contacto'.

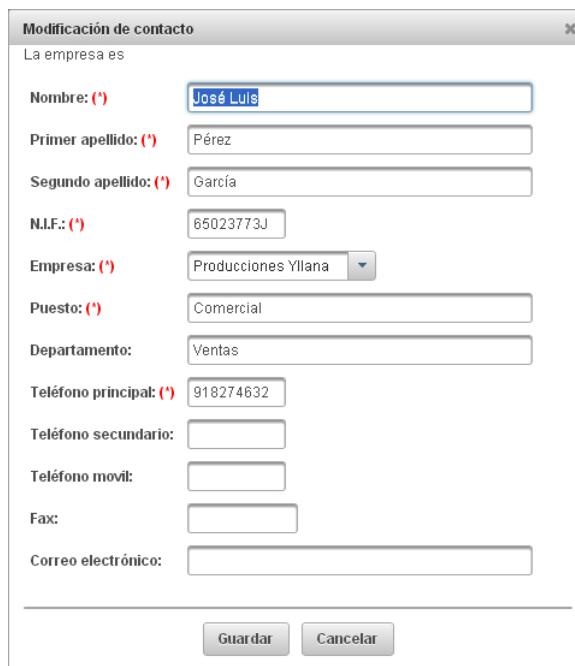
The contact list is displayed in a table with the following columns: 'Nombre y apellidos', 'Empresa, puesto y departamento', 'Teléfonos', 'Fax', 'Correo electrónico', 'Editar contacto', and 'Eliminar contacto'.

Nombre y apellidos	Empresa, puesto y departamento	Teléfonos	Fax	Correo electrónico	Editar contacto	Eliminar contacto
Francisco Pérez Castillo	Producciones Yllana Coordinador	Teléfono principal: 916849382		fpcastillo@produccionesyllana.com		
Javier Alonso García	Rol Lalá Director	Teléfono principal: 918327234				
José Luis Pérez García	Producciones Yllana Comercial (Ventas)	Teléfono principal: 918274632				

At the bottom of the table, there is a pagination bar showing 'Página 1 de 1'.

Figura 8.37. Pantalla de gestión de contactos. Edición y borrado

Si pulsamos sobre el icono de edición de alguno de los contactos, se mostrará el diálogo de edición que podemos ver en la Figura 8.38.



Modificación de contacto

La empresa es

Nombre: (*) José Luis

Primer apellido: (*) Pérez

Segundo apellido: (*) García

N.I.F.: (*) 65023773J

Empresa: (*) Producciones Yllana

Puesto: (*) Comercial

Departamento: Ventas

Teléfono principal: (*) 918274632

Teléfono secundario:

Teléfono móvil:

Fax:

Correo electrónico:

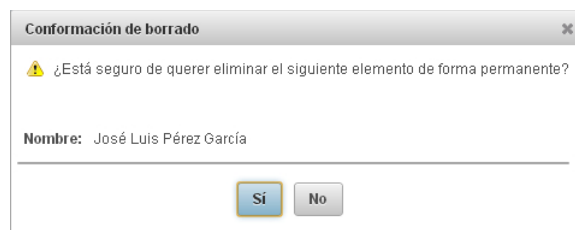
Guardar **Cancelar**

Figura 8.38. Diálogo de edición de contacto

En este formulario se podrán editar todos los campos del contacto, teniendo en cuenta que, en caso de dejar en blanco alguno de los campos obligatorios o de que el N.I.F. ya exista se mostrará el correspondiente mensaje de error al pulsar el botón **Guardar**. En caso contrario, se actualizará el contacto y se reflejarán los cambios en el elemento apropiado del listado de contactos de empresa.

8.2.5.3 Borrado de un contacto

Si el sistema tiene registrado algún contacto, en el listado que se muestra en la ventana de gestión de los mismos aparecerá, junto a cada uno de ellos, un icono de borrado (Figura 8.37). Si pulsamos sobre dicho icono, se mostrará el diálogo de confirmación de borrado del contacto seleccionado que podemos ver en la siguiente figura:



Confirmación de borrado

⚠ ¿Está seguro de querer eliminar el siguiente elemento de forma permanente?

Nombre: José Luis Pérez García

Sí **No**

Figura 8.39. Diálogo de confirmación de borrado de contacto

Este cuadro de diálogo muestra un resumen de la información relativa al contacto a eliminar, y proporciona al usuario dos botones de operación, el botón **Si**, que eliminará el contacto seleccionado, y

	Gestión de espacios y eventos culturales (GEEC) Proyecto de fin de carrera	
---	---	--

el botón **No**, que cancelará la operación y volverá a la pantalla de listado de contactos.

8.2.5.4 Búsqueda o filtro de contactos

La página de gestión de contactos permite realizar búsquedas por el nombre y/o apellidos, N.I.F. o empresa del contacto. Para ello, introduciremos los parámetros en el campo correspondiente del formulario de búsqueda que aparece en la parte superior de la ventana (Figura 8.37) y pulsaremos el botón **Buscar**. Los resultados de la búsqueda aparecerán en el listado de la parte inferior de la ventana, si hay alguno, o en caso contrario se mostrará el correspondiente mensaje informativo.

Para limpiar los parámetros de búsqueda pulsaremos el botón **Limpiar** y volverá a mostrarse el listado de contactos al completo.

8.2.6 Gestión de espacios culturales

Los usuarios Superadministrador o cualquier usuario con rol Editor podrán acceder a la gestión de espacios culturales relativos a su grupo, en caso del Editor, o a cualquier grupo en caso del Superadministrador. Para abrir la pantalla de gestión de espacios culturales habrá que seleccionar el submenú **Eventos** del menú lateral y, dentro del mismo, seleccionar la opción **Gestión de espacios**. Esto mostrará la ventana de operaciones sobre espacios culturales que podemos ver en la Figura 8.40.

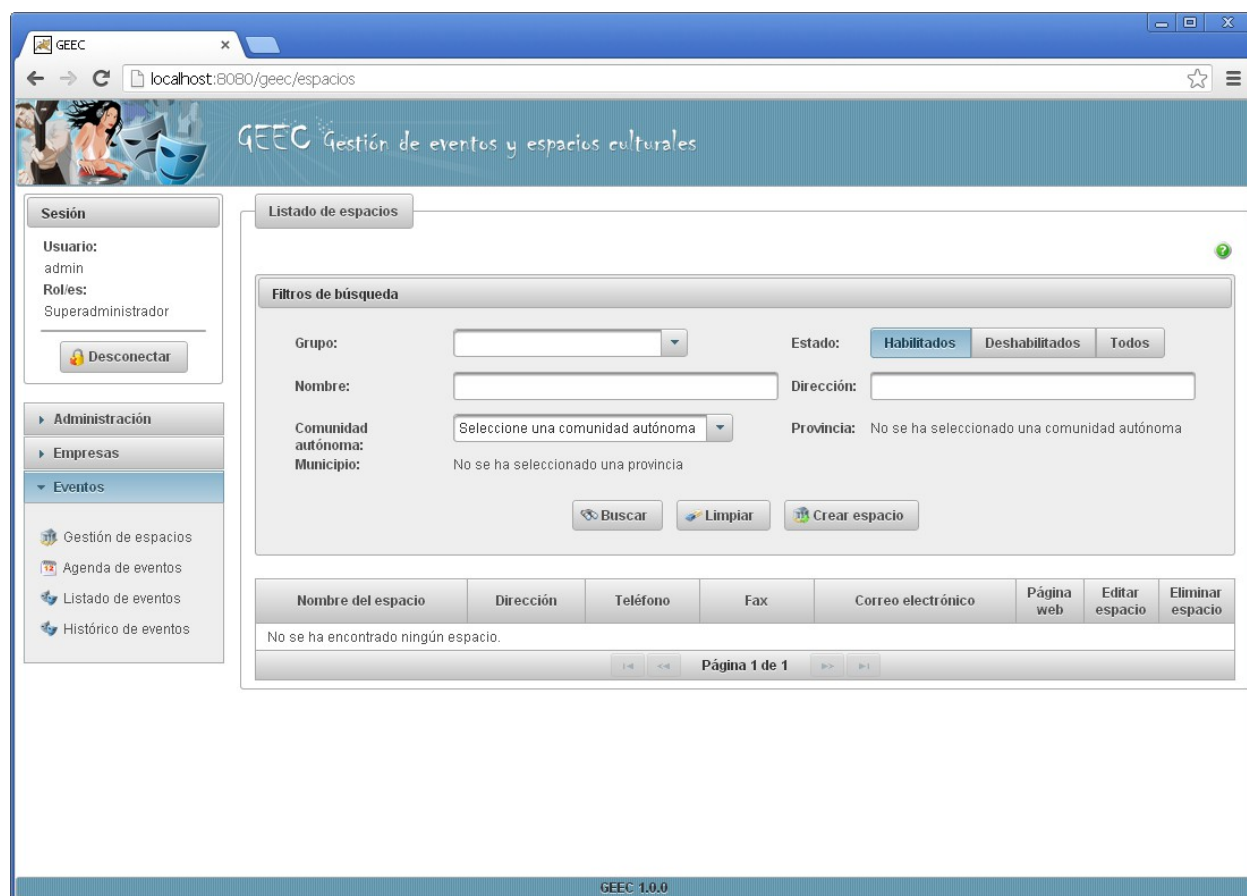


Figura 8.40. Pantalla de gestión de espacios culturales

Sobre esta pantalla se pueden realizar las operaciones de búsqueda de espacios con filtros, búsqueda a la totalidad, creación de un nuevo espacio, edición de un espacio ya existente y borrado de espacios culturales. También se dispondrá de ayuda en pantalla para el usuario pulsando sobre el icono de ayuda que aparece en la esquina superior derecha. Si hacemos clic sobre dicho icono se mostrará el diálogo de información y ayuda de la pantalla, la cual estará disponible para el usuario en todo momento.

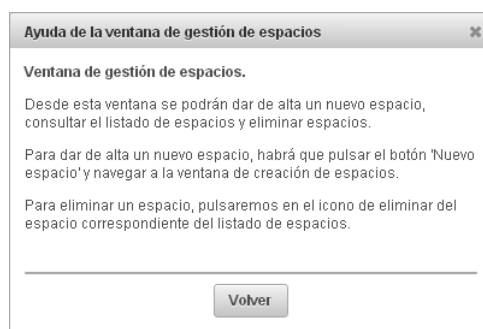
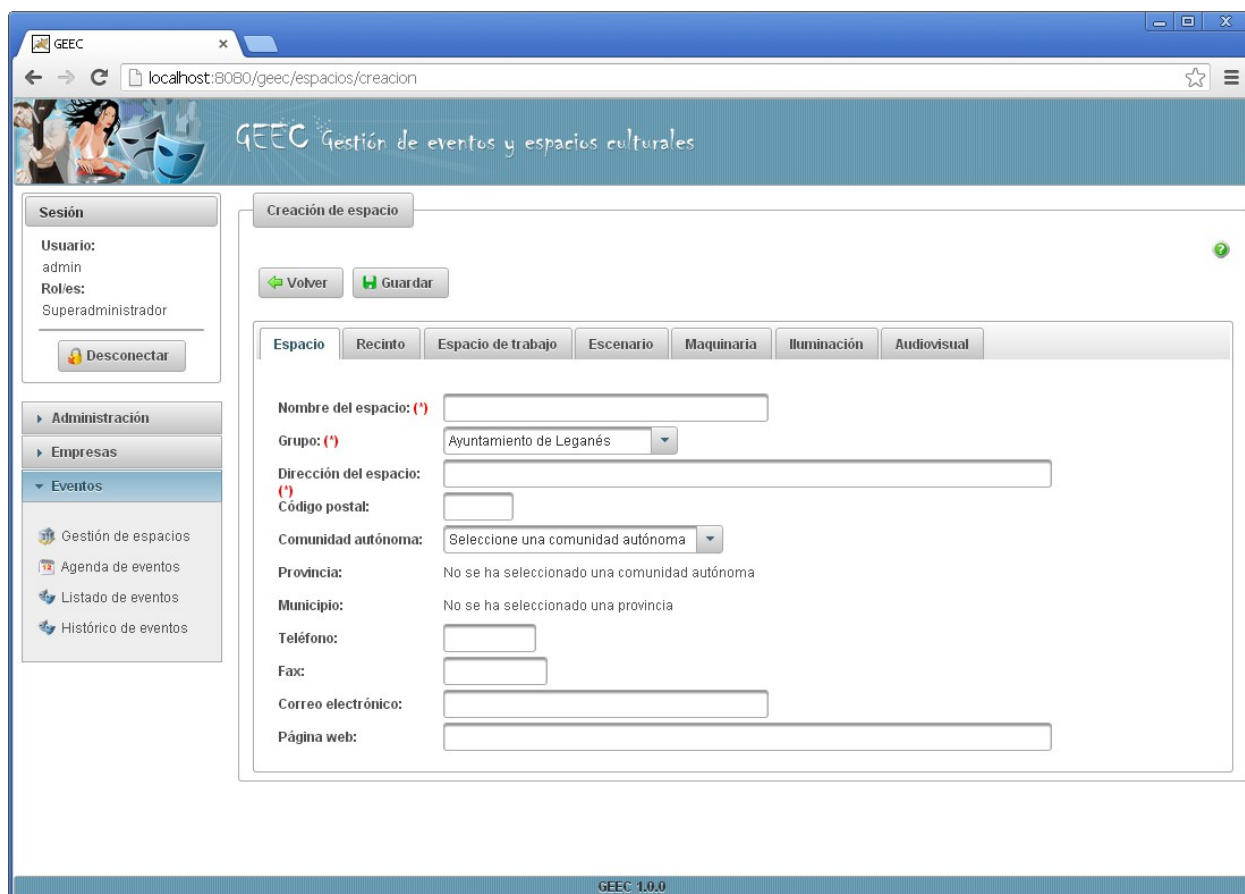


Figura 8.41. Diálogo de ayuda de la pantalla de gestión de espacios culturales

8.2.6.1 Creación de un nuevo espacio cultural

Cualquier usuario podrá crear un nuevo contacto pulsando sobre el botón **Crear espacio** (Figura 8.40), lo cual abrirá la ventana de creación de un nuevo espacio cultural que podemos ver en la siguiente figura:



The screenshot shows the 'GEEC Gestión de eventos y espacios culturales' web application. The browser address bar indicates 'localhost:8080/geec/espacios/creacion'. The page has a sidebar with navigation links: Sesión (Usuario: admin, Roles: Superadministrador, Desconectar), Administración, Empresas, and Eventos (Gestión de espacios, Agenda de eventos, Listado de eventos, Histórico de eventos). The main content area is titled 'Creación de espacio' and features a 'Volver' button and a 'Guardar' button. Below these are tabs for different types of spaces: Espacio, Recinto, Espacio de trabajo, Escenario, Maquinaria, Iluminación, and Audiovisual. The 'Espacio' tab is active, displaying the following form fields:

- Nombre del espacio: (*) [Text input]
- Grupo: (*) [Dropdown menu with 'Ayuntamiento de Leganés' selected]
- Dirección del espacio: [Text input]
- Código postal: (*) [Text input]
- Comunidad autónoma: [Dropdown menu with 'Seleccione una comunidad autónoma' selected]
- Provincia: [Text input with placeholder 'No se ha seleccionado una comunidad autónoma']
- Municipio: [Text input with placeholder 'No se ha seleccionado una provincia']
- Teléfono: [Text input]
- Fax: [Text input]
- Correo electrónico: [Text input]
- Página web: [Text input]

The footer of the application indicates 'GEEC 1.0.0'.

Figura 8.42. Ventana de creación de un nuevo espacio

Esta ventana es bastante compleja, debido a la diversa cantidad de información que se puede almacenar acerca de un espacio donde se ejecuten actividades culturales, por lo cual está dividida en distintas pestañas que agrupan la información según su tipo.

La pestaña más importante y la única que contiene información que es obligatorio rellenar es la pestaña **Espacio** (Figura 8.42). Sobre esta pestaña aparecerán los únicos campos que es obligatorio rellenar sobre un espacio, el nombre del mismo, el grupo al cual pertenece (sólo será editable en caso del Superadministrador, en caso contrario se rellenará con el grupo al cual pertenezca el usuario autenticado) y la dirección del mismo. Opcionalmente, sobre esta pestaña se podrán rellenar el código postal del espacio, la comunidad/provincia/municipio, el teléfono y el fax de contacto, el correo electrónico y la página web del mismo.

El resto de pestañas de agrupación contendrán información técnica del espacio y se agruparán en función del tipo de la misma.

La pestaña **Recinto** permitirá rellenar la información técnica acerca de las dimensiones y la capacidad del recinto (Figura 8.43).

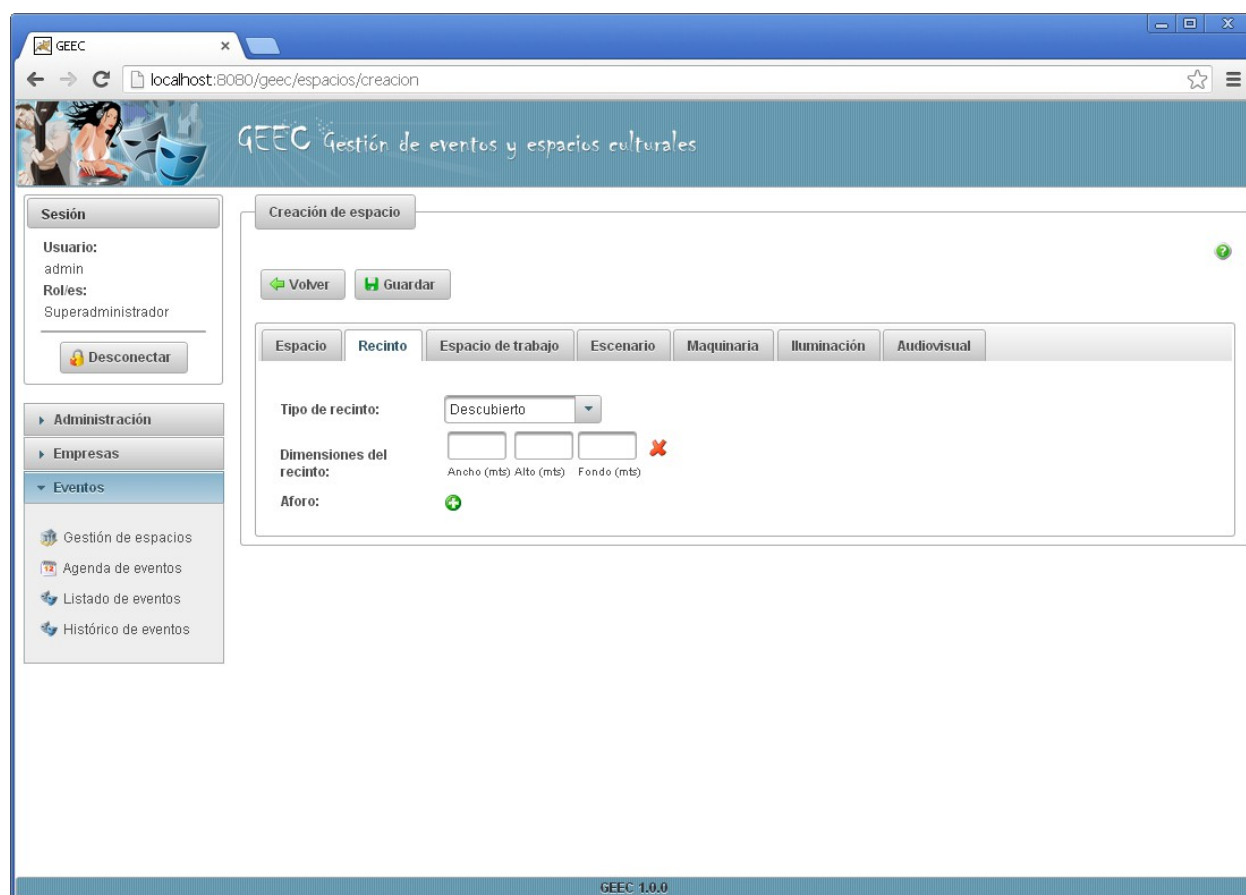
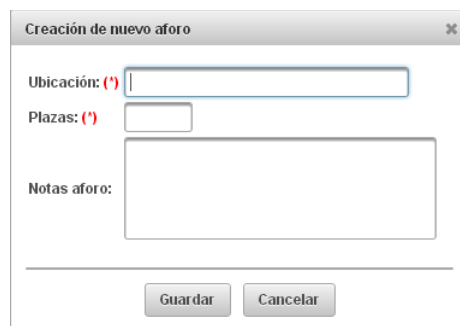


Figura 8.43. Pestaña de información del recinto del espacio

En esta pestaña todos los datos son opcionales. Para añadir las **dimensiones del recinto** pulsaremos sobre el símbolo más (+) y se mostrarán los cuadros de texto para introducir el ancho, alto y fondo del mismo.

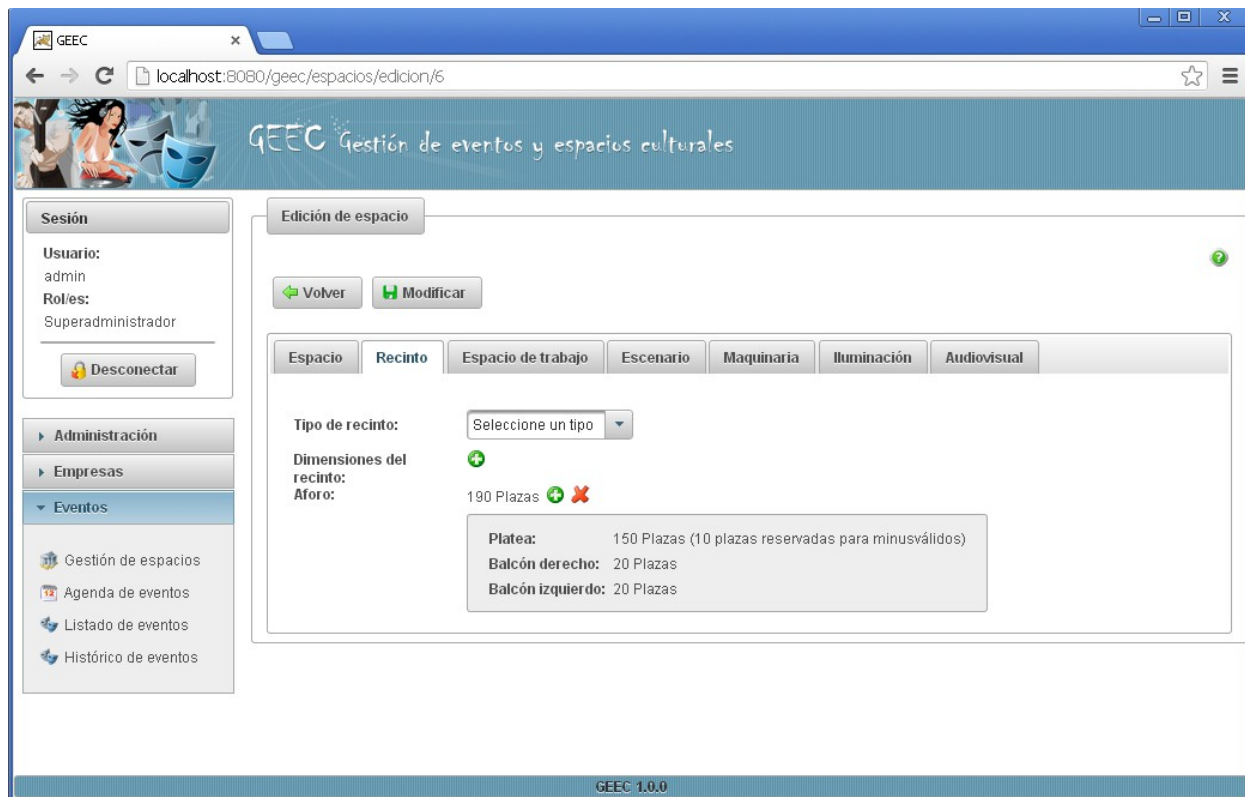
Para eliminar las dimensiones del recinto tendremos que pulsar en el icono (x) que se muestra junto a los campos relativos a las mismas (Figura 8.43).

En el caso del **aforo**, este se puede dividir entre las distintas partes que compongan el recinto (platea, balcón, etc...), por lo que para ir añadiendo cada uno de ellos pulsaremos sucesivamente sobre el símbolo más (+) correspondiente y se abrirá el cuadro de diálogo de inserción de aforo que podemos ver en la siguiente figura:

**Figura 8.44.** Diálogo de inserción de aforo

Para añadir un nuevo aforo, habrá que introducir obligatoriamente los datos de la ubicación y el número de plazas existentes en la misma. Podemos añadir opcionalmente notas sobre el aforo, como butacas reservadas para minusválidos u otros aspectos de interés. Una vez rellenados los datos del espacio de aforo pulsaremos el botón **Guardar** para almacenarlos con el espacio o **Cancelar** si queremos cancelar la inserción. Podemos añadir tantos espacios de aforo como sean necesarios.

Para eliminar un espacio de aforo tendremos que pulsar en el icono (x) que aparece junto al total del aforo y el icono de añadir nuevo aforo (Figura 8.45).

**Figura 8.45.** Pestaña de recinto. Borrado de aforos



Al pulsar sobre el icono de borrado se abrirá el cuadro de diálogo de selección del aforo a eliminar que podemos ver en la siguiente figura:

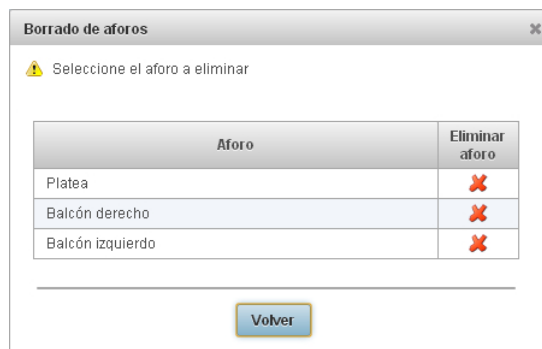


Figura 8.46. Diálogo de selección de aforo a eliminar

Pulsaremos en el icono de eliminar de los aforos que se quieran borrar y una vez terminemos la operación pulsaremos el botón **Volver**.

La pestaña **Espacio de trabajo** permitirá rellenar la información técnica acerca de la distribución u organización del espacio de trabajo del recinto (vestuarios, camerinos, talleres, almacenes, etc...) si dispusiese de los mismos, tal como podemos ver en la siguiente figura:



The screenshot shows a web browser window with the URL `localhost:8080/geec/espacios/creacion`. The page title is "GEEC Gestión de eventos y espacios culturales". On the left, there is a sidebar with a "Sesión" section showing the user as "admin" with the role "Superadministrador" and a "Desconectar" button. Below this are navigation links for "Administración", "Empresas", and "Eventos", with "Eventos" expanded to show "Gestión de espacios", "Agenda de eventos", "Listado de eventos", and "Histórico de eventos". The main content area is titled "Creación de espacio" and contains tabs for "Espacio", "Recinto", "Espacio de trabajo" (which is active), "Escenario", "Maquinaria", "Iluminación", and "Audiovisual". Below the tabs, there are four rows of information, each with a label and a value followed by a green plus icon: "Camerinos: No tiene", "Sala de ensayo: No tiene", "Talleres: No tiene", and "Almacenes: No tiene". At the top of the main area are "Volver" and "Guardar" buttons. The footer of the browser window shows "GEEC 1.0.0".

Figura 8.46. Pestaña de información del espacio de trabajo

En esta pestaña todos los datos son opcionales. Para añadir las distintas opciones habrá que tener en cuenta que puede haber varios elementos para cada uno de ellos, por lo que para ir añadiendo cada uno de ellos pulsaremos sucesivamente sobre el símbolo más (+) correspondiente.

Para añadir un nuevo elemento de **camerinos** se mostrará el diálogo de la Figura 8.47:

The dialog box is titled "Creación de nuevo camerino". It contains four fields: "Género: (*)" with a dropdown menu set to "Masculino", "Tipo: (*)" with a dropdown menu set to "Individual", "Nº de camerinos: (*)" with a text input field, and "Notas camerinos:" with a larger text area. At the bottom, there are "Guardar" and "Cancelar" buttons.

Figura 8.46. Diálogo de inserción de camerino

En este diálogo será obligatorio seleccionar el género del vestuario (masculino, femenino o mixto), el tipo (individual o colectivo) y el número de camerinos de este tipo de que dispone el espacio. Opcionalmente se pueden añadir notas sobre el o los camerinos que se consideren importantes (como si dispone de ducha, si se puede reutilizar como camerino, etc...).

Una vez creados los camerinos deseados, se mostrará junto al número total de los mismos y el icono de creación de nuevo camerino, el icono de borrado de camerinos (Figura 8.47).



Figura 8.47. Pestaña de espacio de trabajo. Borrado de elementos

Para borrar uno o todos los camerinos pulsaremos sobre dicho icono y se abrirá el diálogo de selección de los camerinos a eliminar (Figura 8.48).

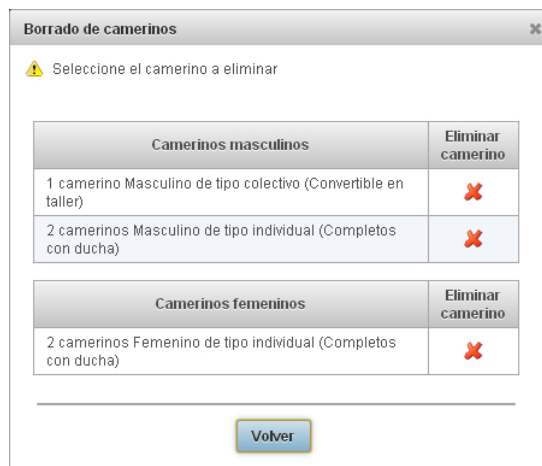


Figura 8.48. Diálogo de borrado de camerinos

Pulsaremos sobre el icono de borrado de los camerinos que se deseen eliminar y una vez terminado pulsaremos el botón **Volver**.

Para añadir una **sala de ensayo** se mostrará el diálogo de la Figura 8.49:

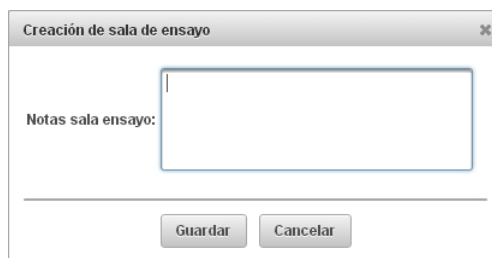


Figura 8.49. Diálogo de inserción de sala de ensayo

El espacio podrá añadir o disponer de una sala de ensayo, para la cual se pueden añadir notas sobre la misma que se consideren importantes. Para eliminar la sala de ensayo pulsaremos sobre el icono de borrado que se mostrará junto a la sala de ensayo añadida al espacio (Figura 8.47).

Para añadir **talleres** a un espacio habrá que pulsar el símbolo (+) que se encuentra junto a la etiqueta correspondiente. Se mostrará el cuadro de diálogo de la Figura 8.50, en el cual podremos introducir el número de talleres de que dispone el espacio y las notas que se consideren necesarias acerca de los mismos.

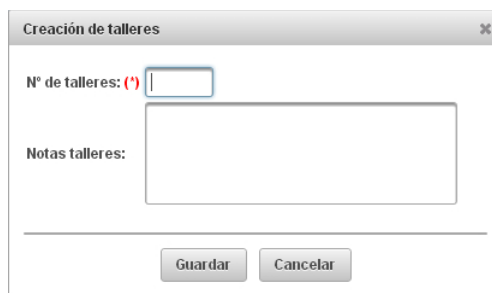


Figura 8.50. Diálogo de inserción de talleres

Una vez introducido el número de talleres que tiene el espacio y las notas sobre los mismos que se consideren importantes pulsaremos el botón **Guardar** para añadir los talleres o el botón **Cancelar** para descartar la inserción. Si se han añadido talleres al espacio, podremos pulsar sobre el icono de borrado que aparecerá junto a los mismos para eliminarlos de la información del espacio.

El último elemento a añadir en esta pestaña son los almacenes del espacio. Para añadirlos pulsaremos en el icono de insertar almacén que aparece junto a la etiqueta para mostrar el diálogo de inserción de almacenes de la Figura 8.51.

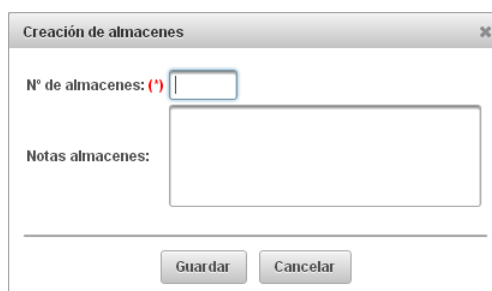


Figura 8.51. Diálogo de inserción de almacenes

Una vez introducido el número de almacenes y las notas sobre los mismos que se consideren importantes pulsaremos el botón **Guardar** para añadirlos al espacio o el botón **Cancelar** para descartar la inserción. Si se han añadido almacenes, podremos pulsar sobre el icono de borrado que aparecerá junto a los mismos para eliminarlos de la información del espacio.

La pestaña **Escenario** permitirá rellenar la información técnica acerca del tipo, las dimensiones y equipamiento de que dispone el escenario del espacio cultural si lo tuviese (Figura 8.52).

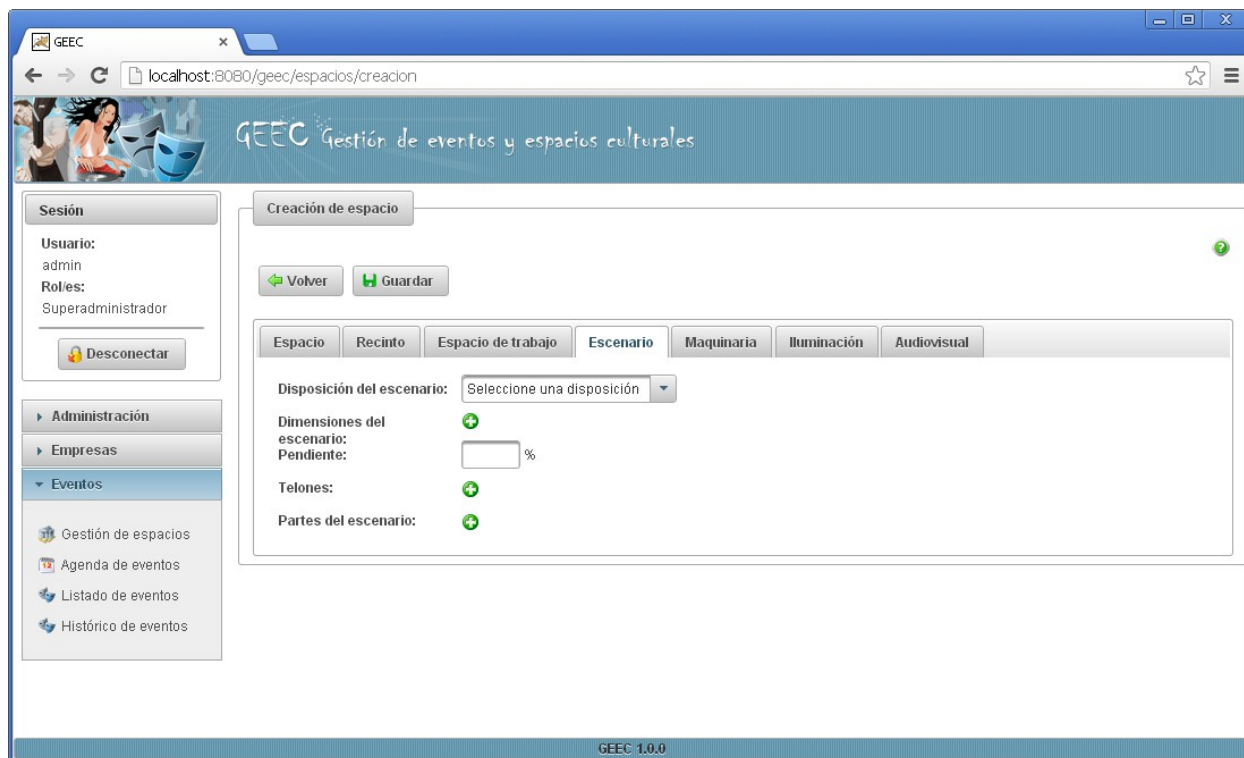


Figura 8.52. Pestaña de información del escenario del espacio

En esta pestaña podemos seleccionar la **Disposición del escenario** del espacio entre los tipos parametrizados en el sistema (Italiana, Isabelina, Circular o Anfiteatro).

También se podrán añadir las dimensiones del mismo pulsando el botón de insertar dimensiones que aparece junto a la etiqueta **Dimensiones del escenario**, el cual mostrará los elementos de entrada para los valores del ancho y el fondo del mismo. Si se ha seleccionado introducir dimensiones para el escenario se mostrará junto a los elementos de entrada el icono de eliminar, que permitirá borrar los valores introducidos.

El siguiente valor que se puede introducir en la información del escenario son los **Telones**. Para ello pulsaremos sobre el botón de añadir telón, mostrándose el diálogo de inserción de nuevo telón que podemos ver en la siguiente figura:

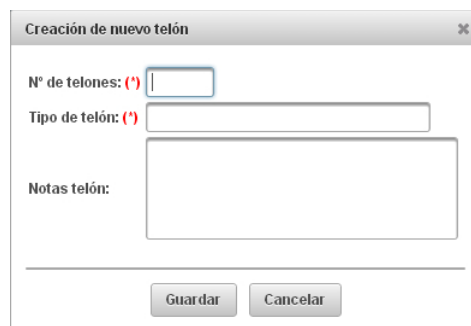


Figura 8.53. Diálogo de creación de nuevo telón

En este diálogo se podrá introducir el número de telones del tipo deseado, el tipo y las notas sobre el telón que se crean convenientes. Los campos número y tipo de telón son obligatorios y si no se introducen se mostrará el correspondiente mensaje de error. Una vez configurado el telón, pulsaremos el botón **Guardar** para insertarlo o **Cancelar** para anular la inserción. Al insertar algún elemento de tipo telón para el espacio, se mostrará junto al icono de insertar nuevo telón el icono de borrado como podemos observar en la Figura 8.54.

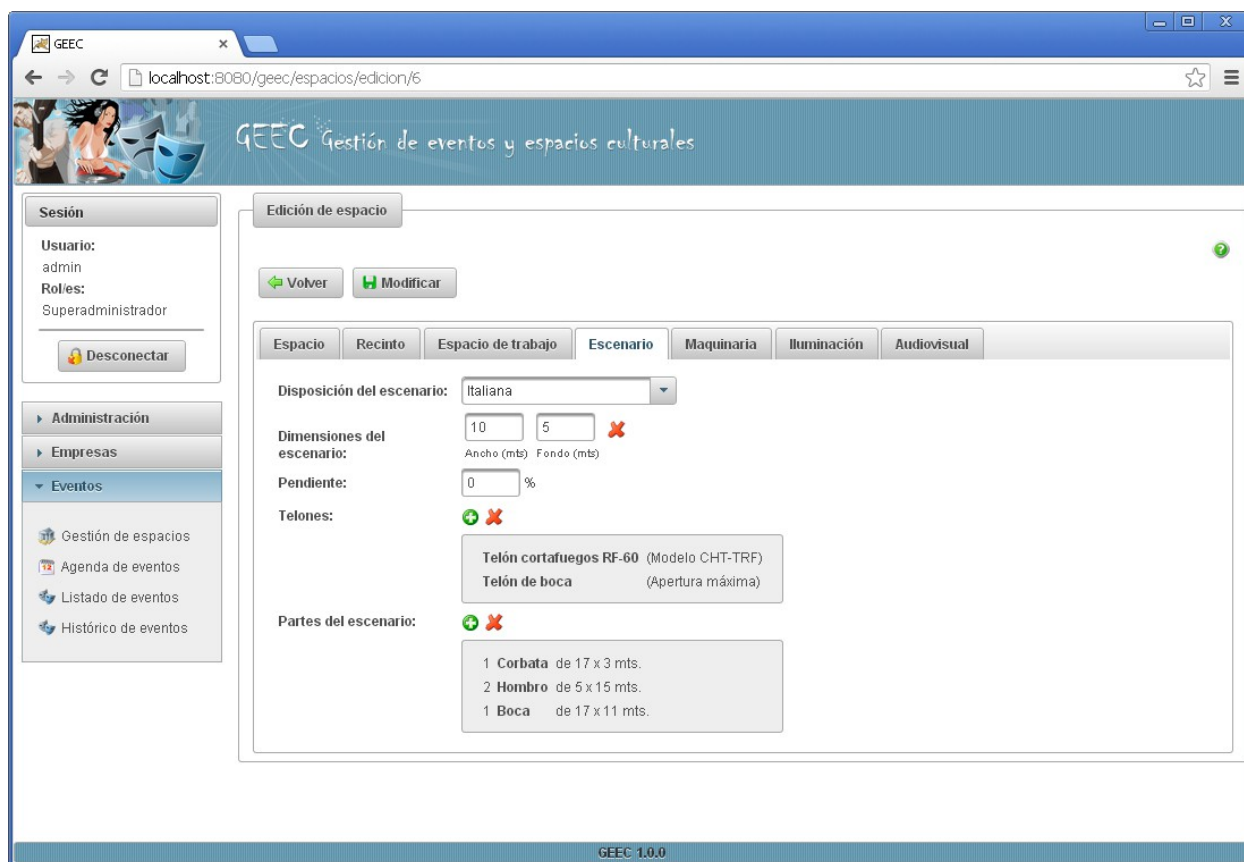


Figura 8.54. Pestaña de escenario. Iconos de borrado de elementos

Para eliminar un telón, pulsaremos sobre el icono correspondiente y se abrirá el diálogo de borrado de telones (Figura 8.55). En este diálogo, pulsaremos sobre el icono de borrado de cada uno de los telones que queramos eliminar, y una vez finalizada la acción pulsaremos el botón **Volver** para regresar a la pestaña de escenario donde veremos reflejados los cambios realizados y los correspondientes mensajes de información.

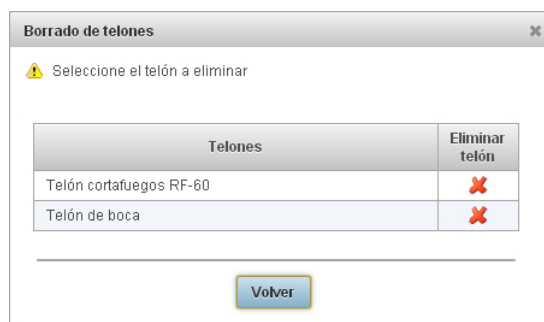


Figura 8.55. Diálogo de borrado de telones

El último elemento que se puede configurar en la pestaña de escenario son las distintas partes y dimensiones que lo componen. Para añadir una **Parte de escenario** pulsaremos sobre el icono de inserción que aparece junto a la etiqueta, y se abrirá el diálogo de inserción de nueva parte de escenario (Figura 8.56).

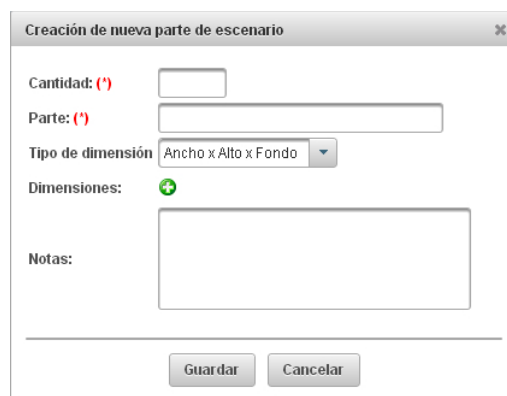


Figura 8.56. Diálogo de inserción de parte de escenario

Las partes de escenario constarán de información sobre la cantidad, el tipo de parte de que se trata, el tipo de dimensión a introducir (en caso de que se vaya a introducir), las dimensiones en relación con el tipo anterior y notas importantes sobre las mismas. Los únicos campos obligatorios serán la cantidad de la parte y el tipo. Para añadir la dimensión de la parte seleccionaremos en primer lugar el tipo de dimensión (alto x ancho, alto x ancho x fondo, etc...) y pulsaremos en el botón de inserción que se muestra junto a la etiqueta **Dimensiones** para que aparezcan los campos relativos a las mismas. Una vez mostrados los campos, podremos pulsar el icono de eliminar dimensión para borrarla.



Cuando se hayan introducido los campos obligatorios y los opcionales que se consideren necesarios, pulsaremos el botón **Guardar** para insertar la parte en el espacio o **Cancelar** para descartar la inserción. Si el espacio tiene alguna parte de escenario insertada, se mostrará junto al icono de inserción de partes el icono de borrado (Figura 8.54). Si pulsamos el icono de borrado se mostrará el diálogo de borrado de partes de escenario que podemos ver en la siguiente figura.

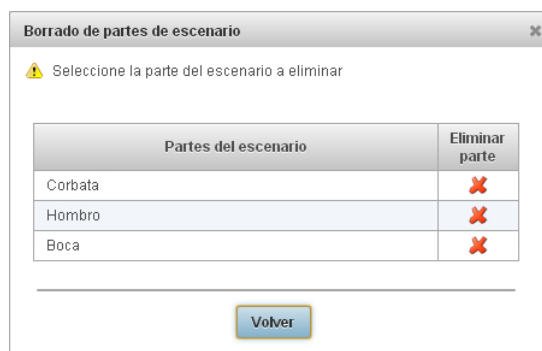


Figura 8.57. Diálogo de borrado de partes de escenario

En este diálogo podremos pulsar el icono de borrado de las distintas partes que se quieran eliminar, y una vez terminada la operación pulsaremos el botón **Volver** para ver reflejados los cambios en la pestaña de escenario junto con los correspondientes mensajes de información.

La pestaña **Maquinaria** permitirá rellenar la información técnica acerca de la maquinaria de que dispone el espacio cultural (Figura 8.58).

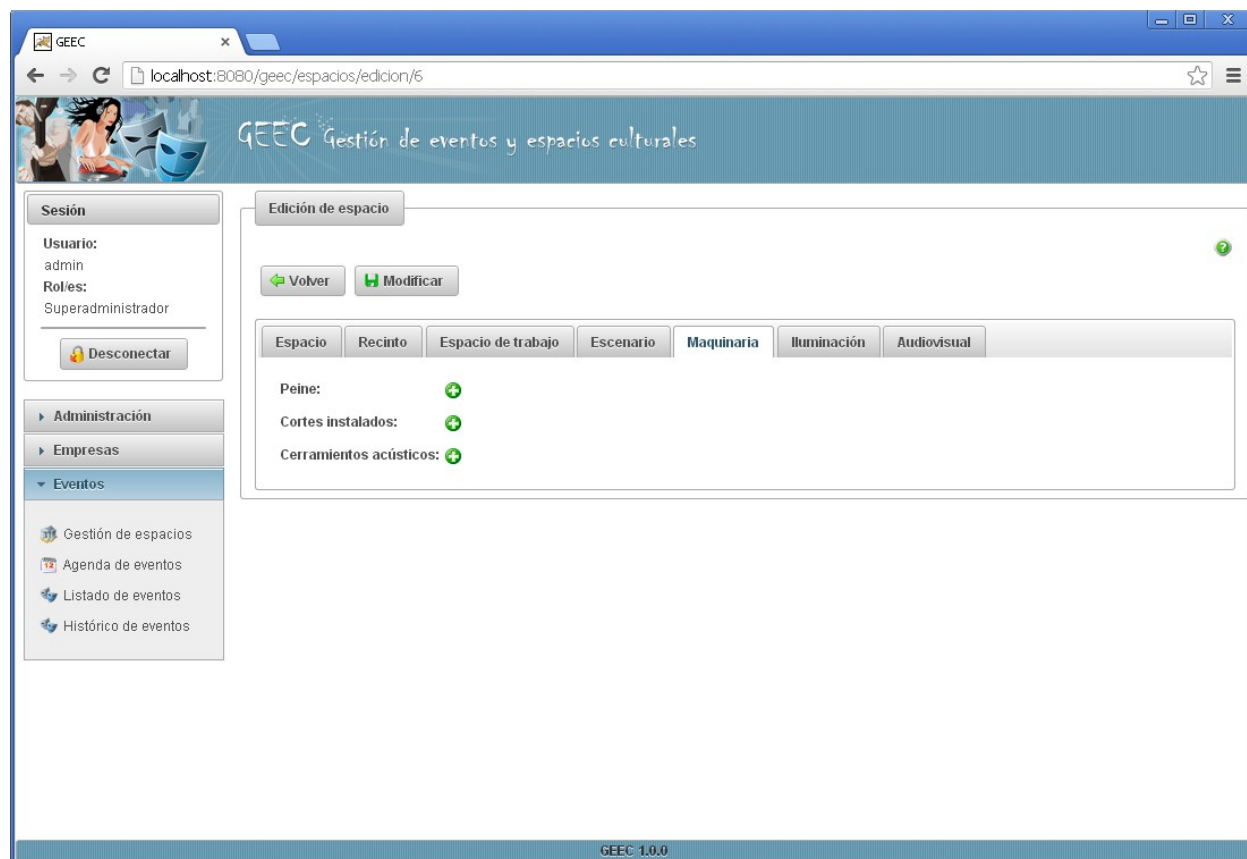


Figura 8.58. Pestaña de información de la maquinaria del espacio

En esta pestaña podremos introducir información sobre el peine, los cortes instalados y el cerramiento acústico del espacio si los tuviese. Si el espacio tiene **Peine** pulsaremos el icono de inserción para añadir la información sobre el mismo en el diálogo de inserción de peine (Figura 8.59).

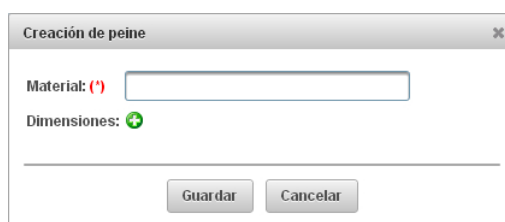


Figura 8.59. Diálogo de inserción de peine

Debemos introducir obligatoriamente el material con que está fabricado el peine y de forma opcional las dimensiones del mismo. Una vez introducidos los datos, pulsaremos el botón **Guardar** para insertar el peine o **Cancelar** para descartar la inserción.

Si se ha introducido información sobre el peine del espacio, en la pestaña, junto a la misma, se mostrará el icono de borrado de dicha información (Figura 8.60).

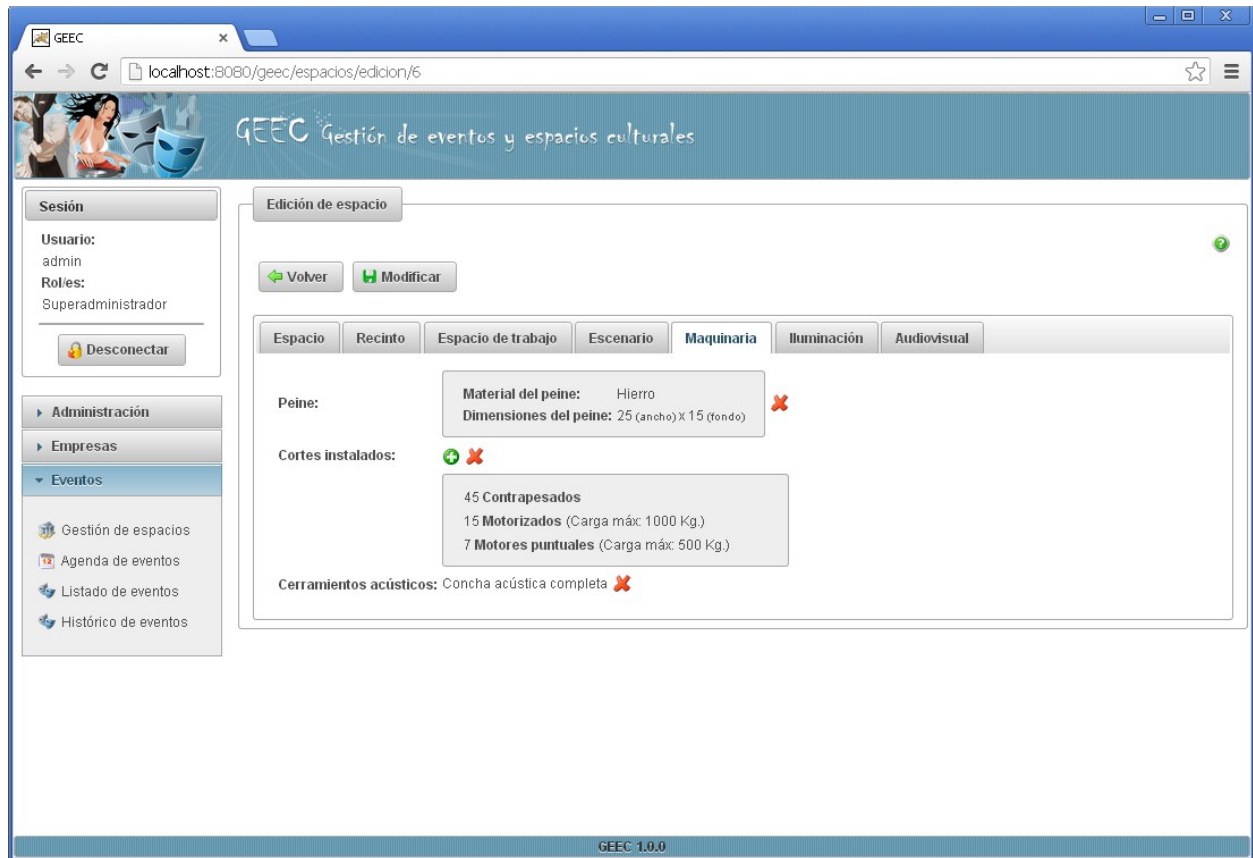


Figura 8.60. Pestaña de maquinaria. Elementos de borrado

Para eliminar la información sobre el peine, pulsaremos el icono de borrado y los cambios se reflejarán en la pestaña de maquinaria junto con el correspondiente mensaje de información.

Para insertar cortes en el espacio, pulsaremos en el icono de inserción que aparece junto a la etiqueta **Cortes instalados**. Esto mostrará el diálogo de inserción de corte de la Figura 8.61.

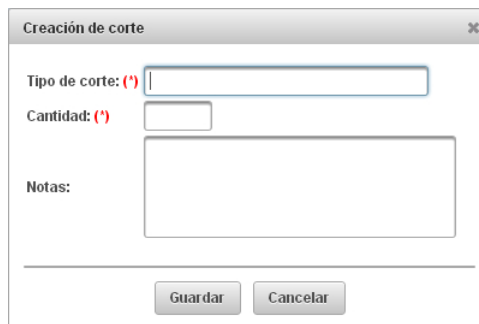


Figura 8.61. Diálogo de inserción de corte

Para insertar un nuevo corte será necesario introducir su tipo y la cantidad del mismo. Opcionalmente se

podrán introducir notas sobre el corte a insertar. Pulsaremos el botón **Guardar** para insertar el corte o **Cancelar** para descartar la inserción.

Si el espacio tiene información sobre cortes, se mostrará junto al icono de inserción un icono de borrado (Figura 8.60). Si pulsamos en el icono de borrado se mostrará el diálogo de borrado de cortes del espacio de la Figura 8.62.

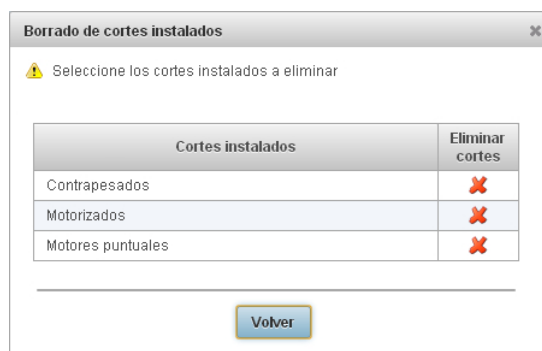


Figura 8.62. Diálogo de borrado de cortes

En este diálogo podremos pulsar el icono de borrado de los cortes que se quieran eliminar, y una vez terminada la operación pulsaremos el botón **Volver** para ver reflejados los cambios en la pestaña de maquinaria junto con los correspondientes mensajes de información.

El último elemento que podemos introducir en la pestaña de maquinaria es si el espacio tiene cerramiento acústico. Para ello, pulsaremos en el icono de inserción que aparece junto a la etiqueta **Cerramientos acústicos**, y se mostrará el diálogo de inserción de cerramiento (Figura 8.63).

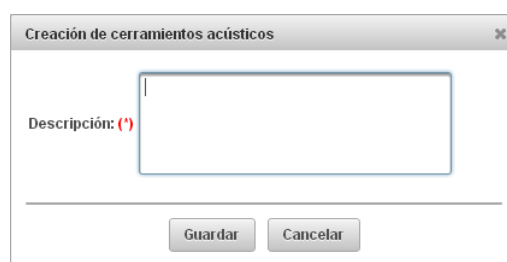


Figura 8.63. Diálogo de inserción de cerramiento

Será obligatorio introducir la descripción del cerramiento del espacio. Para eliminar el mismo una vez introducido, pulsaremos en el icono de borrado que aparece junto al mismo (Figura 8.60).

La pestaña **Iluminación** permitirá rellenar la información técnica acerca de los componentes de iluminación de que dispone el espacio (Figura 8.64).

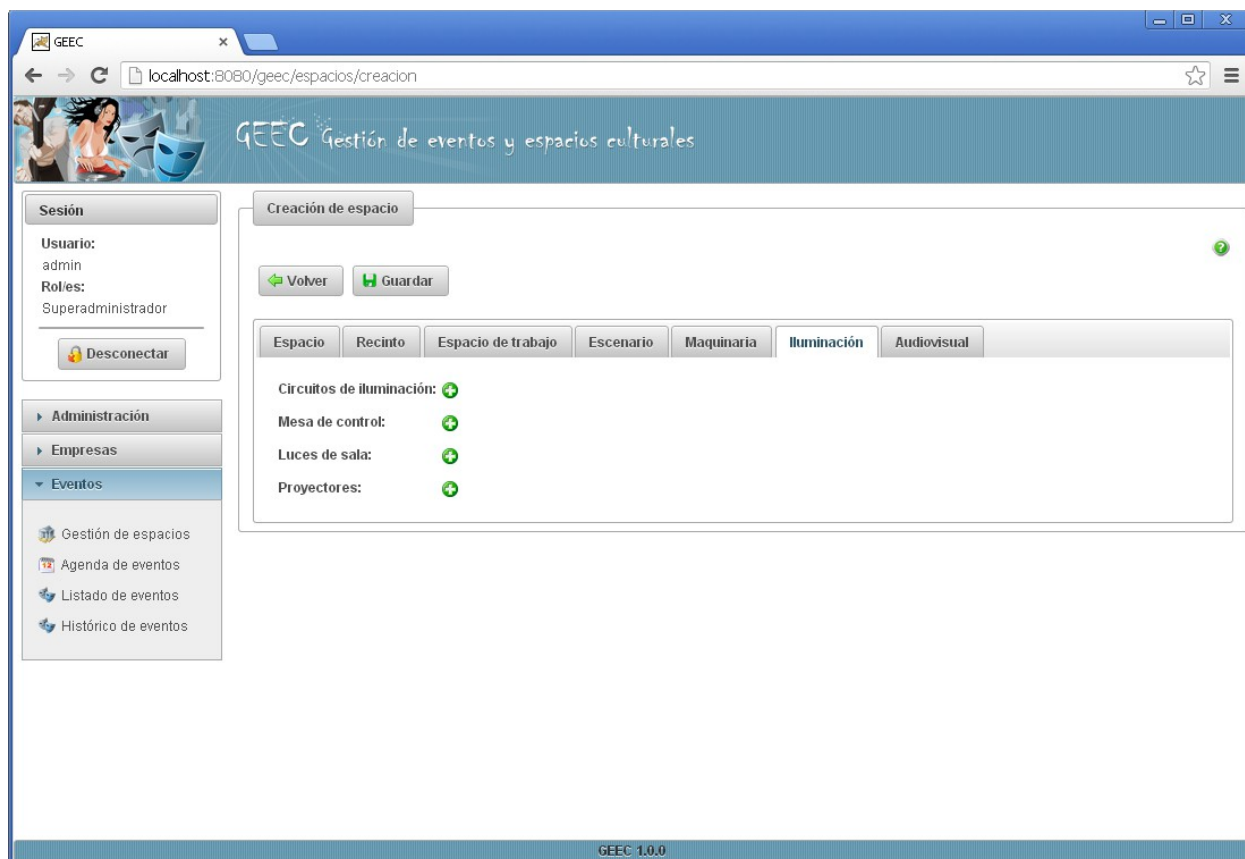


Figura 8.64. Pestaña de información de la iluminación del espacio

En esta pestaña podremos introducir información sobre los circuitos de iluminación, la mesa de control, las luces de la sala y los proyectores de luz si los tuviese. Si el espacio tiene circuitos pulsaremos el icono de inserción que aparece junto a la etiqueta **Circuitos de iluminación** para añadir tantos como sea necesario mediante el diálogo de inserción de circuitos que se muestra en la Figura 8.65.

Figura 8.65. Diálogo de inserción de circuito

Para insertar un nuevo circuito será necesario introducir su tipo y la cantidad del mismo. Opcionalmente se podrán introducir notas sobre el circuito a insertar. Pulsaremos el botón **Guardar** para insertar el

circuito o **Cancelar** para descartar la inserción.

Si el espacio tiene información sobre circuitos instalados, se mostrará junto al icono de inserción un icono de borrado (Figura 8.66).

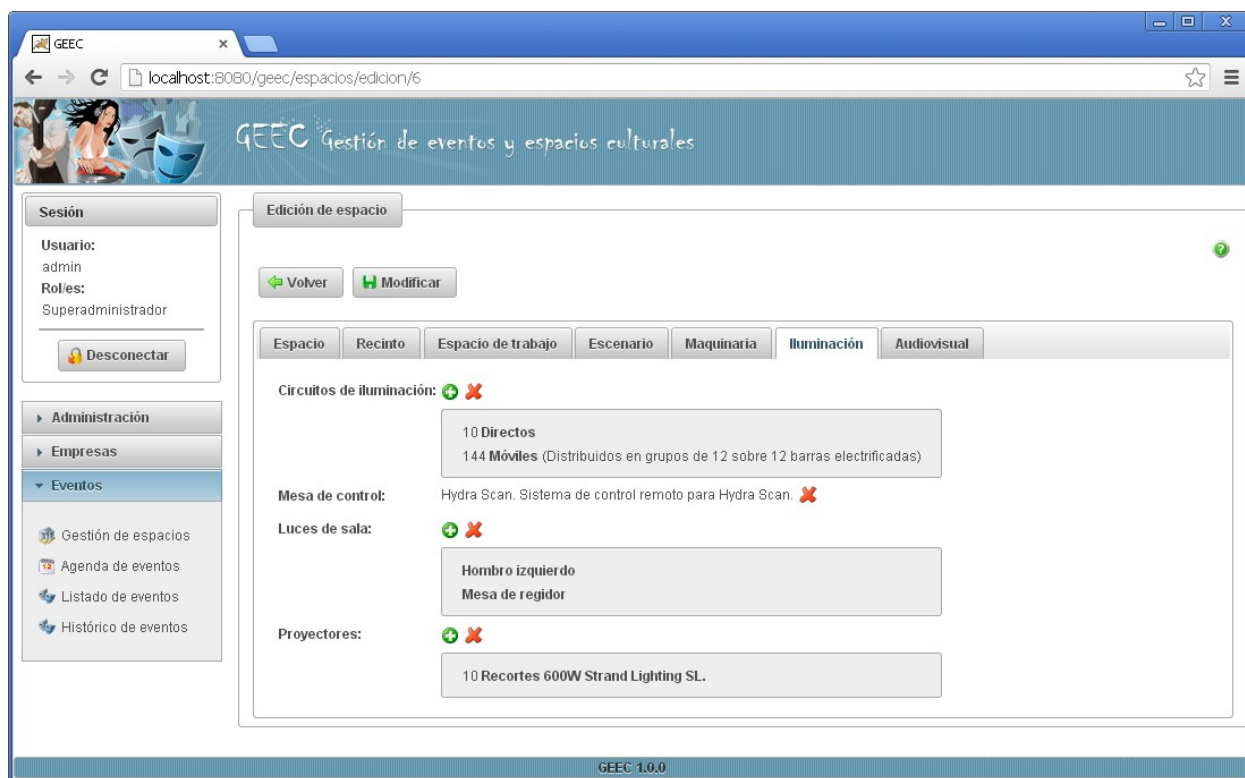


Figura 8.66. Pestaña de información de la iluminación del espacio. Borrado de elementos

Si pulsamos en el icono de borrado de circuito, se mostrará el cuadro de diálogo que podemos ver en la siguiente figura.

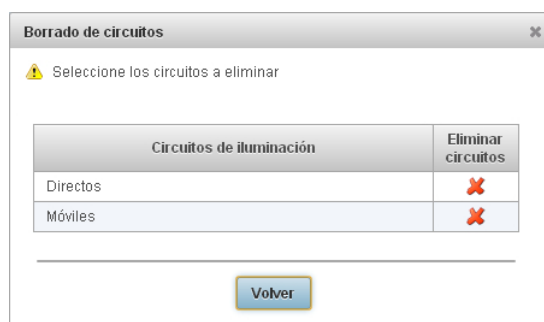
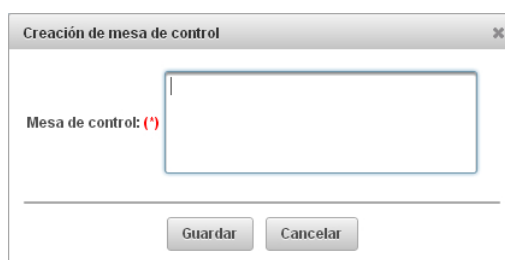


Figura 8.67. Diálogo de borrado de circuitos

En este diálogo podremos pulsar el icono de borrado de los circuitos que se quieran eliminar, y una vez terminada la operación pulsaremos el botón **Volver** para ver reflejados los cambios en la pestaña de

iluminación junto con los correspondientes mensajes de información.

Si el espacio dispone de mesa de control de iluminación, pulsaremos en el icono de inserción que aparece junto a la etiqueta **Mesa de control** y se mostrará el diálogo que aparece en la siguiente figura, que permite la inserción de la mesa.



Creación de mesa de control

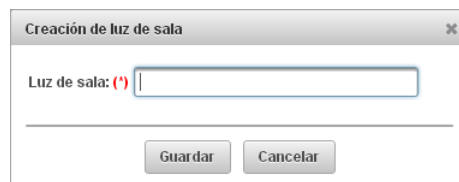
Mesa de control: (*)

Guardar Cancelar

Figura 8.68. Diálogo de inserción de mesa de control

Para insertar la mesa de control de iluminación sólo será necesario introducir la descripción o notas importantes sobre la misma. Una vez añadida la mesa de control, podemos pulsar en el botón de eliminar que aparecerá junto a la información sobre misma (Figura 8.66) para borrarla.

En esta pestaña también se pueden insertar las posiciones de las luces de la sala. Para ello pulsaremos en el botón de inserción que aparece junto a la etiqueta **Luces de sala** (Figura 8.64). Se mostrará el diálogo de inserción de luz de sala que se muestra en la siguiente figura.



Creación de luz de sala

Luz de sala: (*)

Guardar Cancelar

Figura 8.69. Diálogo de inserción de luz de sala

En este diálogo será obligatorio introducir la luz de sala o posición donde se encuentre situada dicha luz. Podrán introducirse tantas luces de sala como sea necesario. Una vez insertada alguna luz de sala, se mostrará junto al icono de inserción el icono de borrado de luces de sala (Figura 8.66) que nos permitirá borrar una o varias de ellas. Si pulsamos sobre el icono de borrado se mostrará el diálogo de la Figura 8.70.

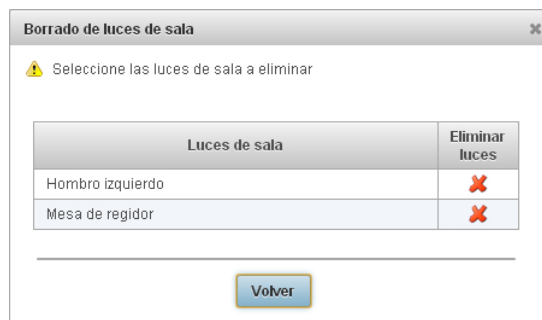


Figura 8.70. Diálogo de borrado de luces de sala

En este diálogo podremos pulsar el icono de borrado de las luces de sala que se quieran eliminar, y una vez terminada la operación pulsaremos el botón **Volver** para ver reflejados los cambios en la pestaña de iluminación junto con los correspondientes mensajes de información.

El último elemento que podemos añadir en esta pestaña son los proyectores de luz. Para insertar un proyector de luz pulsaremos el icono de inserción que aparece junto a la etiqueta **Proyectores** (Figura 8.64). Esto mostrará el diálogo de inserción de proyector que se muestra en la siguiente figura.

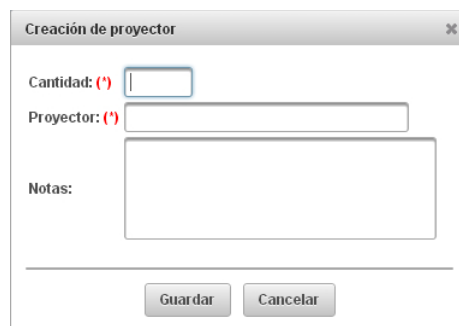


Figura 8.71. Diálogo de inserción de proyector

Para insertar un nuevo proyector será necesario introducir su tipo y la cantidad del mismo a insertar. Opcionalmente se podrán introducir notas sobre el proyector a insertar. Pulsaremos el botón **Guardar** para insertar el proyector o **Cancelar** para descartar la inserción.

Si el espacio tiene información sobre proyectores, se mostrará junto al icono de inserción un icono de borrado (Figura 8.66). Al pulsar sobre el mismo se mostrará el diálogo de borrado de proyectores en el cual podremos pulsar el icono de borrado de los proyectores que se quieran eliminar, y una vez terminada la operación pulsaremos el botón **Volver** para ver reflejados los cambios en la pestaña de iluminación junto con los correspondientes mensajes de información.



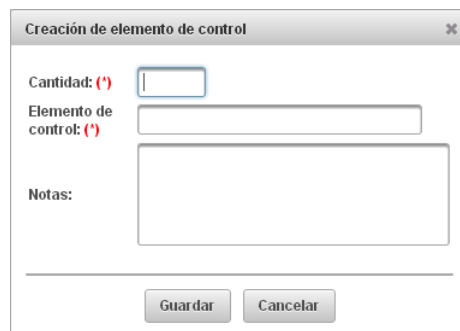
Figura 8.72. Diálogo de borrado de proyectores

La última pestaña es la de información **Audiovisual**, la cual permitirá rellenar la información técnica acerca de los componentes audiovisuales de que dispone el espacio (Figura 8.73).



Figura 8.73. Pestaña de información audiovisual del espacio

En esta pestaña podremos introducir información sobre los elementos de control, los equipos de refuerzo sólido, los altavoces y la microfonía de que dispone el espacio. Si el espacio tiene elementos de control audiovisual, pulsaremos el icono de inserción que aparece junto a la etiqueta **Elementos de control** para añadir tantos como sea necesario mediante el diálogo de inserción de los mismos que se muestra en la Figura 8.74.



Creación de elemento de control

Cantidad: (*)

Elemento de control: (*)

Notas:

Figura 8.74. Diálogo de inserción de elemento de control

Para insertar un nuevo elemento de control será necesario introducir su cantidad y el elemento de que se trata. Opcionalmente se podrán introducir notas sobre el elemento de control a insertar. Pulsaremos el botón **Guardar** para insertarlo o **Cancelar** para descartar la inserción.

Si el espacio tiene información sobre elementos de control audiovisual, se mostrará junto al icono de inserción un icono de borrado (Figura 8.75).



GEEC Gestión de eventos y espacios culturales

Sesión

Usuario: admin

Roles: Superadministrador

Administración

Empresas

Eventos

Gestión de espacios

Agenda de eventos

Listado de eventos

Histórico de eventos

Edición de espacio

Espacio Recinto Espacio de trabajo Escenario Maquinaria Iluminación **Audiovisual**

Elementos de control: + -

1 Mesa de sonido Midas legend (40 canales, 10 VCA-12 Aux - 8 grupo...)

1 Ecualizador gráfico Samson E-62 (Multiefectos)

Equipos de refuerzo sólido: + -

2 Reproductores de CD

1 Pletina doble de cassette Tascam 302 M K II

Altavoces: + -

1 Caja acústica subgraves cañón AWCS II

14 Cajas acústicas compacta Bose W 802 serie II

Micrófonos: + -

4 Micros crown LM 300

4 Micros samsung S11

GEEC 1.0.0

Figura 8.75. Pestaña de información audiovisual del espacio. Borrado de elementos

Si pulsamos en el icono de borrado de elemento de control, se mostrará el cuadro de diálogo que podemos ver en la siguiente figura.

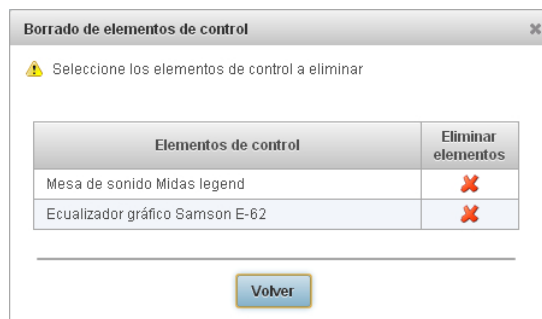


Figura 8.76. Diálogo de borrado de elementos de control

En este diálogo podremos pulsar el icono de borrado de los elementos de control que se quieran eliminar, y una vez terminada la operación pulsaremos el botón **Volver** para ver reflejados los cambios en la pestaña de información audiovisual junto con los correspondientes mensajes de información.

Para insertar un equipo de refuerzo sólido pulsaremos el icono de inserción que aparece junto a la etiqueta **Equipos de refuerzo sólido** (Figura 8.73). Esto mostrará el diálogo de inserción del elemento que se muestra en la siguiente figura.

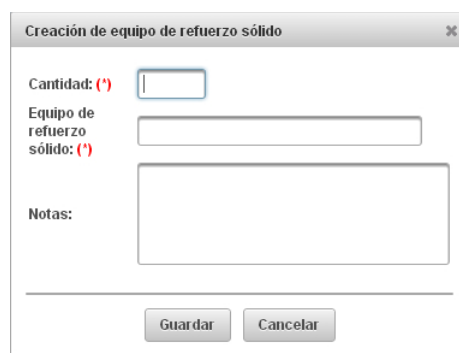


Figura 8.77. Diálogo de inserción de equipo de refuerzo sólido

Para insertar un nuevo equipo será necesario introducir su cantidad y el modelo o tipo del mismo a insertar. Opcionalmente se podrán introducir notas sobre el equipo de refuerzo a insertar. Pulsaremos el botón **Guardar** para insertarlo o **Cancelar** para descartar la inserción.

Si el espacio tiene información sobre equipos de refuerzo sólido en la pestaña de información audiovisual, se mostrará junto al icono de inserción un icono de borrado (Figura 8.75). Al pulsar sobre el mismo se mostrará el diálogo de borrado de equipos de refuerzo sólido en el cual podremos pulsar el icono de borrado de los equipos que se quieran eliminar, y una vez terminada la operación pulsaremos el botón **Volver** para ver reflejados los cambios en la pestaña de información audiovisual del espacio junto

con los correspondientes mensajes de información.

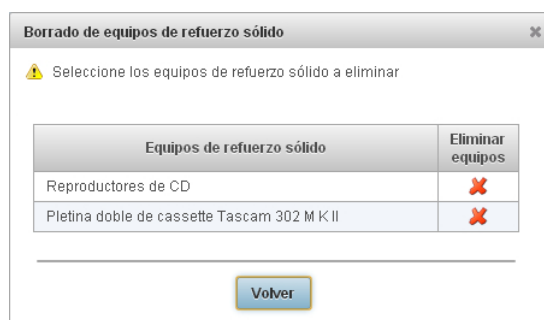


Figura 8.78. Diálogo de borrado de equipos de refuerzo sólido

Para insertar altavoces pulsaremos el icono de inserción que aparece junto a la etiqueta **Altavoces** (Figura 8.73), lo cual hará que se muestre el diálogo de inserción del elemento que se muestra en la siguiente figura.

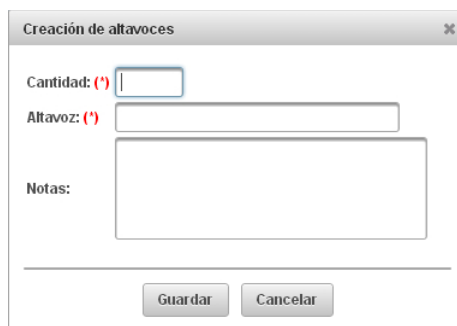


Figura 8.79. Diálogo de inserción de altavoces

Para insertar un nuevo conjunto de altavoces será necesario introducir obligatoriamente su cantidad y el modelo del mismo. Opcionalmente se podrán introducir notas sobre el altavoz. Pulsaremos el botón **Guardar** para insertarlo o **Cancelar** para descartar la inserción.

Si el espacio tiene información sobre altavoces en la pestaña de información audiovisual, se mostrará junto al icono de inserción un icono de borrado (Figura 8.75). Al pulsar sobre el mismo se mostrará el diálogo de borrado de altavoces (Figura 8.80) en el cual podremos pulsar el icono de borrado de los altavoces que se quieran eliminar, y una vez terminada la operación pulsaremos el botón **Volver** para ver reflejados los cambios en la pestaña de información audiovisual del espacio junto con los correspondientes mensajes de información.

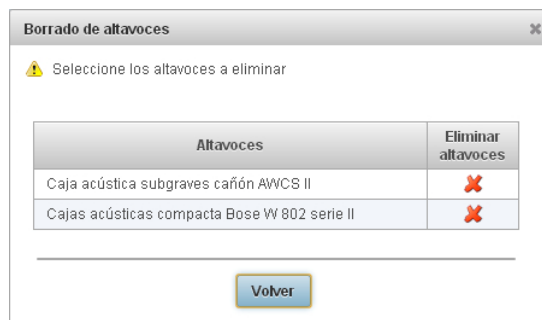


Figura 8.80. Diálogo de borrado de altavoces

El último elemento que podemos añadir en esta pestaña son los elementos de microfonía. Para insertar un conjunto de micrófonos pulsaremos el icono de inserción que aparece junto a la etiqueta **Micrófonos** (Figura 8.73). Esto mostrará el diálogo de inserción de micrófonos que se muestra en la siguiente figura.

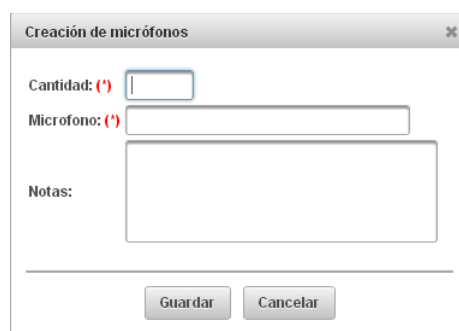


Figura 8.81. Diálogo de inserción de micrófonos

Para insertar un nuevo conjunto de micrófonos será necesario introducir su modelo y la cantidad de los mismos a insertar. Opcionalmente se podrán introducir notas sobre el elemento de microfonía a insertar. Pulsaremos el botón **Guardar** para insertarlo o **Cancelar** para descartar la inserción.

Si el espacio tiene información sobre micrófonos, se mostrará junto al icono de inserción un icono de borrado (Figura 8.75). Al pulsar sobre el mismo se mostrará el diálogo de borrado de micrófonos en el cual podremos pulsar el icono de borrado de los mismos que se quieran eliminar, y una vez terminada la operación pulsaremos el botón **Volver** para ver reflejados los cambios en la pestaña de información audiovisual junto con los correspondientes mensajes de información.

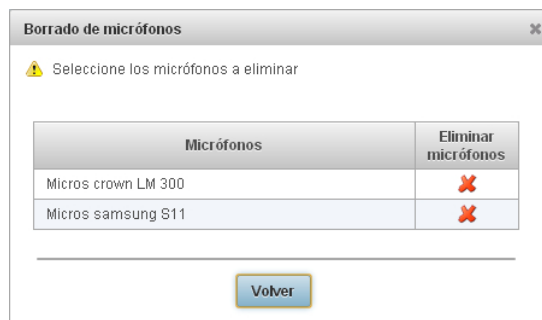


Figura 8.82. Diálogo de borrado de micrófonos

Una vez rellenados los datos deseados, pulsaremos el botón **Guardar** y grabaremos en el sistema el nuevo espacio, siempre y cuando se hayan rellenado al menos los campos obligatorios, en caso contrario se mostrarán los correspondientes mensajes de error. Pulsaremos el botón **Cancelar** si queremos cancelar la creación del espacio. Una vez guardado el espacio cultural, los cambios se reflejarán en el listado de contactos de la ventana principal.

8.2.6.2 Edición de un espacio cultural

Si el sistema tiene registrado algún espacio, en el listado de los mismos que se muestra en la ventana de gestión de espacios aparecerá, junto a cada uno de ellos, un icono de edición (Figura 8.83).

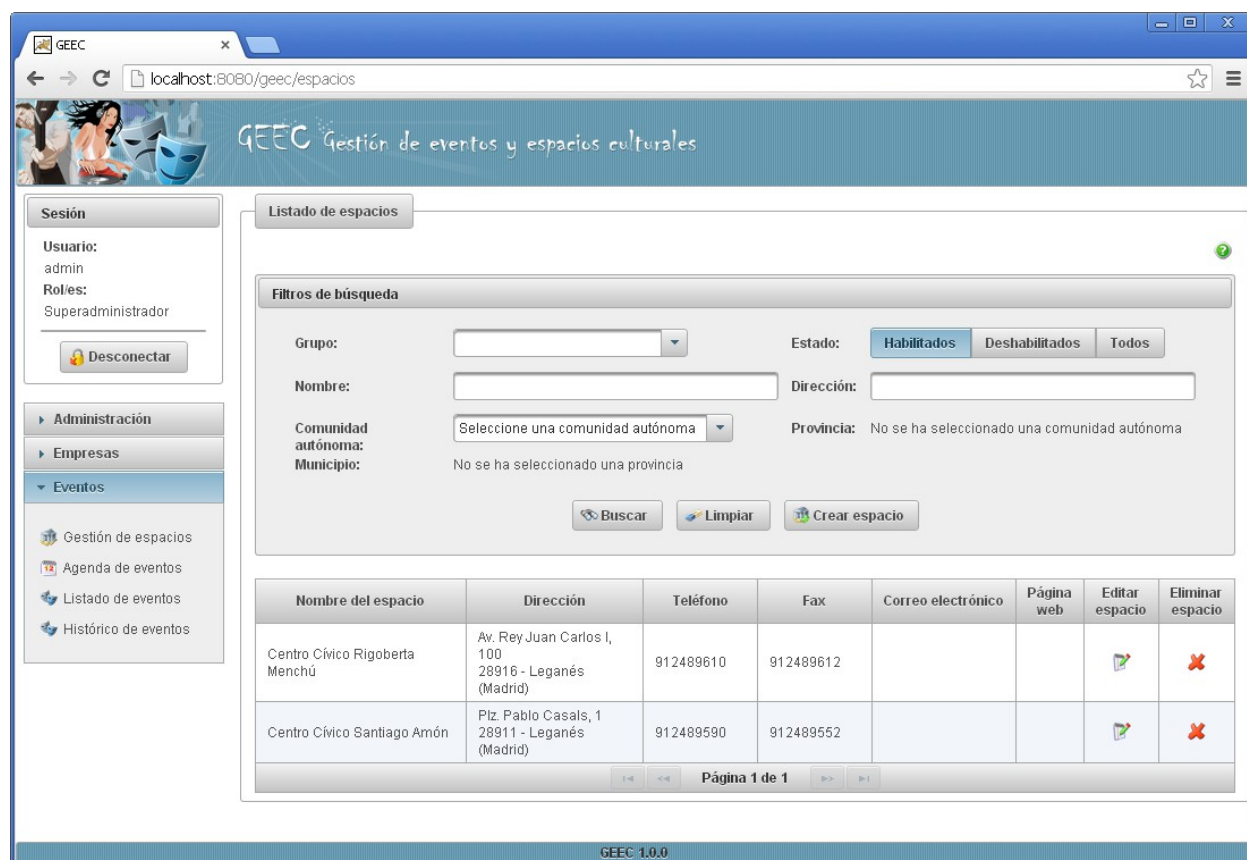


Figura 8.83. Pantalla de gestión de espacios. Edición y borrado

Si pulsamos sobre el icono de edición de alguno de los espacios, navegaremos a la ventana de edición del mismo. Esta ventana dispondrá de las mismas pestañas, opciones y restricciones que la ventana de creación del espacio.

Una vez modificados los datos pulsaremos el botón **Modificar** actualizándose los datos del mismo y mostrando los mensajes de información o de error en la parte superior de la ventana.

8.2.6.3 Borrado de un espacio cultural

Si el sistema tiene registrado algún espacio cultural, en el listado que se muestra en la ventana de gestión de los mismos aparecerá, junto a cada uno de ellos, un icono de borrado (Figura 8.83). Si pulsamos sobre dicho icono, se mostrará el diálogo de confirmación de borrado del espacio seleccionado que podemos ver en la siguiente figura:

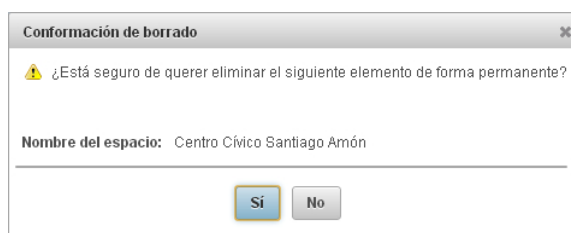


Figura 8.84. Diálogo de confirmación de borrado de espacio

Este cuadro de diálogo muestra el nombre del espacio a eliminar, y proporciona al usuario dos botones de operación, el botón **Si**, que eliminará el espacio seleccionado, y el botón **No**, que cancelará la operación y volverá a la pantalla de listado de espacios culturales.

8.2.6.4 Búsqueda o filtro de espacios

La página de gestión de espacios culturales permite realizar búsquedas por el nombre, la dirección, la comunidad/provincia/municipio, el grupo al que pertenece el espacio y sobre el estado en que se encuentra, es decir, si se encuentra activo (Habilitado), inactivo (Deshabilitado) o sobre todos los espacios independientemente del estado en que se encuentren. Para ello, introduciremos los parámetros en el campo correspondiente del formulario de búsqueda que aparece en la parte superior de la ventana (Figura 8.83) y pulsaremos el botón **Buscar**. Los resultados de la búsqueda aparecerán en el listado de la parte inferior de la ventana, si hay alguno, o en caso contrario se mostrará el correspondiente mensaje informativo.

Para limpiar los parámetros de búsqueda pulsaremos el botón **Limpiar** y volverá a mostrarse el listado de espacios al completo.

8.2.7 Agenda de eventos

La agenda de eventos culturales será la pantalla de entrada en la aplicación. A ella podrán acceder todos los usuarios para consultar los eventos programados, los cuales se mostrarán en modo de agenda paginados de semana en semana con las flechas de navegación que se muestran en la parte superior derecha de la ventana (Figura 8.85).

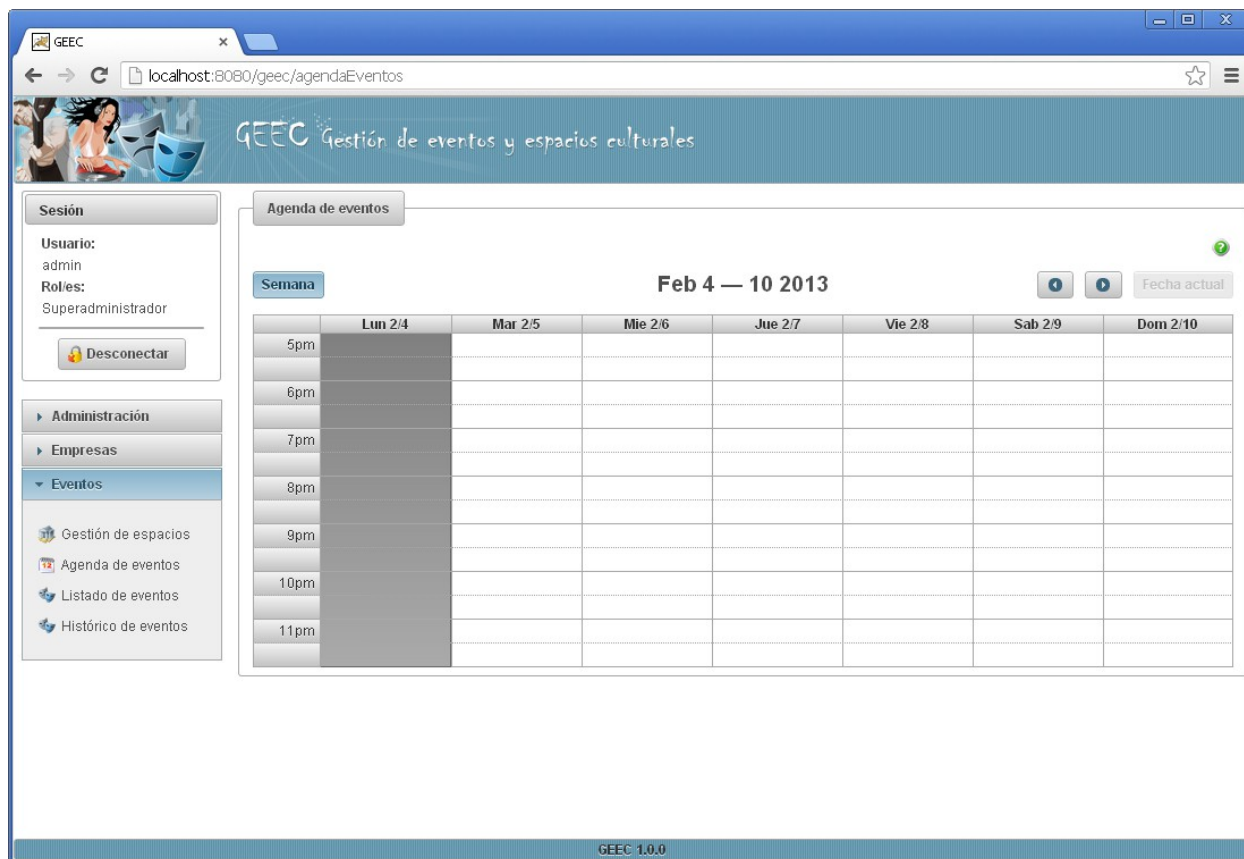


Figura 8.85. Agenda de eventos culturales

Los usuarios Superadministrador o con rol **Editor** podrán utilizar esta ventana para crear o modificar eventos que se hayan programado.

Además, los eventos mostrados en la agenda tendrán cuatro colores representativos:

- Los eventos de color NARANJA son los eventos que han sido creados pero están pendientes de su revisión para aceptarlos o rechazarlos.
- Los eventos de color VERDE son los eventos que han sido aceptados y están a la espera de su ejecución.
- Los eventos de color ROJO son los eventos que han sido rechazados y que están a la espera de su modificación para pasar de nuevo al estado de pendientes de revisión.
- Por último, los eventos de color GRIS son los que ya han sido ejecutados o que estando rechazados ha pasado su fecha de ejecución.

Los eventos en estado pendiente cuya fecha de ejecución sea inferior a 15 días desde la fecha actual pasarán automáticamente al estado rechazado.

Los eventos aceptados cuya fecha de ejecución sea igual a la fecha actual pasarán automáticamente al estado ejecutado.

8.2.7.1 Creación de evento sobre la agenda

Para crear un nuevo evento cultural sobre la agenda de eventos, el usuario deberá hacer clic sobre una casilla de hora en una fecha superior a 15 días a partir de la fecha actual (la cual se marcará en la agenda sobreada en color gris), ya que este es el rango de margen para la validación o rechazo de eventos.

Al pulsar sobre una casilla de hora se mostrará el cuadro de diálogo de creación de evento que podemos ver en la Figura 8.86.

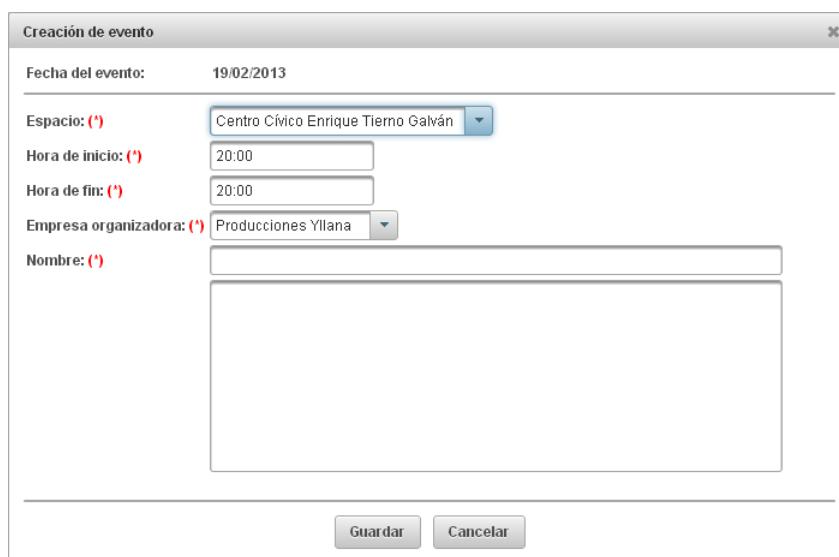


Figura 8.86. Diálogo de creación de evento

El diálogo mostrará en la parte superior la fecha en la cual se va a crear el evento. Será obligatorio seleccionar el espacio donde se va a desarrollar el evento, la hora de inicio y finalización del mismo, la empresa organizadora y el nombre del evento. También es posible introducir de forma opcional una descripción o sinopsis del mismo.

Una vez introducidos los datos pulsaremos el botón **Guardar** para programar el evento o **Cancelar** para volver a la agenda sin introducir ningún cambio.

Una vez programado un evento, los cambios se reflejarán inmediatamente en la agenda, tal como se puede observar en la siguiente figura.

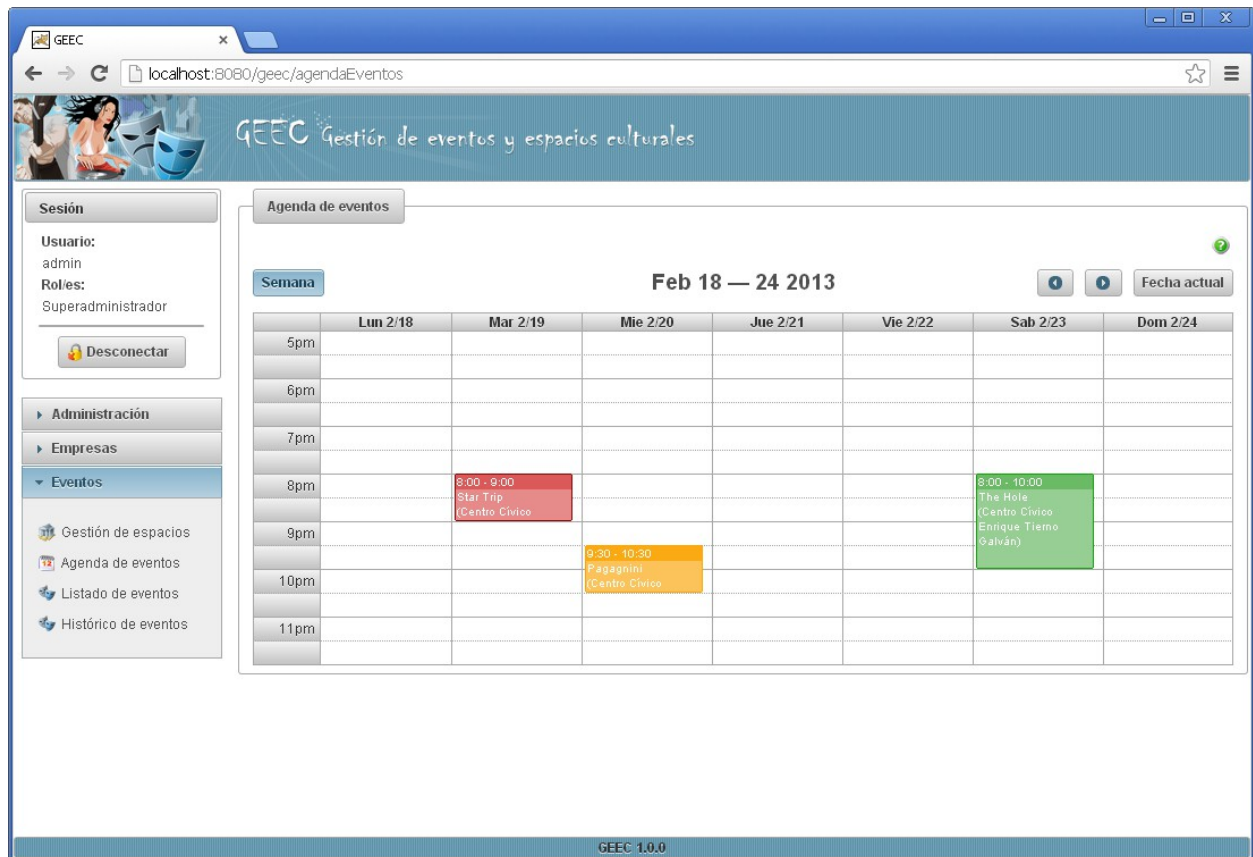


Figura 8.87. Agenda de eventos con eventos en distintos estados

8.2.7.2 Modificación de evento sobre la agenda

El usuario Superadministrador y los usuarios con rol de Editor podrán editar los eventos dados de alta en el sistema desde la agenda de eventos. Para ello haremos clic con el ratón sobre el evento que queramos editar, abriendo así el diálogo de modificación de evento que se muestra en la siguiente figura.



Modificación de evento

Espacio: (*) Centro Cívico Enrique Tierno Galván

Fecha del evento: (*) 19/02/2013

Hora de inicio: (*) 20:00

Hora de fin: (*) 21:00

Empresa organizadora: (*) Producciones Yllana

Nombre: (*) Star Trip

Star Trip, espectáculo maquinado por Yllana, nos adentra en el alucinante y absurdo mundo de la ciencia ficción y el espacio. Todo ello con su característico humor negro, su lenguaje universal y sin palabras.

Los protagonistas de Star Trip son cuatro intrépidos astronautas que llevan 50 años hibernando dentro de una nave espacial que vaga por el espacio, y cuya misión es encontrar vida inteligente en el universo para, una vez hallada, cargársela.

El espectáculo comienza con el despertar de nuestra audaz tripulación. Su lucha por dominar una nave que tiene vida propia, el laborioso día a día en ella, y los diferentes

Modificar **Cancelar**

Figura 8.88. Diálogo de modificación de evento

Sobre este diálogo podremos modificar cualquier campo que queramos respetando la obligatoriedad de los mismos. Una vez realizados los cambios pulsaremos el botón **Modificar** para guardar los cambios o **Cancelar** para descartarlos. Estos cambios se reflejarán en la agenda de eventos junto con los correspondientes mensajes de información.

8.2.8 Listado de eventos

Todos los usuarios del sistema, independientemente del rol que desempeñen, podrán acceder al listado de eventos para consultar el listado de los mismos. En el caso de los usuarios con rol **Editor** sólo podrán utilizarlo en modo de consulta.



The screenshot shows the GEEC web application interface. The browser address bar indicates the URL is localhost:8080/geec/listadoEventos. The page title is 'GEEC Gestión de eventos y espacios culturales'. The sidebar on the left contains a 'Sesión' section with user information (Usuario: jvidal, Grupo: Ayuntamiento de Leganés, Roles: Programador, Editor) and a 'Desconectar' button. Below this is a menu with 'Empresas' and 'Eventos' (selected). The 'Eventos' menu includes links for 'Gestión de espacios', 'Agenda de eventos', 'Listado de eventos', and 'Histórico de eventos'. The main content area is titled 'Listado de eventos' and features a 'Filtros de búsqueda' section with dropdown menus for 'Espacio', 'Empresa', 'Nombre', and 'Estado', along with 'Buscar' and 'Limpiar' buttons. Below the filters is a table of events.

Espacio	Título	Fecha	Horario	Estado
Centro Cívico Enrique Tierno Galván	The Hole (Producciones Yllana)	23/02/2013	Desde: 07:00 Hasta: 09:00	Aceptado
Centro Cívico Enrique Tierno Galván	Pagagnini (Producciones Yllana)	20/02/2013	Desde: 08:30 Hasta: 09:30	Pendiente
Centro Cívico Enrique Tierno Galván	Star Trip (Producciones Yllana)	19/02/2013	Desde: 07:00 Hasta: 08:00	Rechazado

At the bottom of the table, there is a pagination control showing 'Página 1 de 1'.

Figura 8.89. Listado de eventos del Editor. Modo consulta

En el caso de los usuarios con rol **Autorizador** o del usuario Superadministrador podrán utilizar el listado de eventos para consultarlos y para autorizar o rechazar un evento de forma manual, mostrándose los iconos de operación correspondientes en sendas columnas junto a cada uno de los eventos (Figura 8.90). Para aceptar un evento, únicamente tendremos que pulsar en el icono de **Aceptar evento** del evento correspondiente. Para rechazar un evento tendremos que pulsar en el icono de **Rechazar evento** del evento correspondiente.



GEEC Gestión de eventos y espacios culturales

Sesión
Usuario: admin
Roles: Superadministrador
Desconectar

Listado de eventos

Filtros de búsqueda

Grupo:
Espacio: Nombre:
Empresa: Estado:
Buscar Limpiar

Espacio	Título	Fecha	Horario	Estado	Aceptar evento	Rechazar evento
Centro Cívico Enrique Tierno Galván	The Hole (Producciones Yllana)	23/02/2013	Desde: 07:00 Hasta: 09:00	Aceptado	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Centro Cívico Enrique Tierno Galván	Pagagnini (Producciones Yllana)	20/02/2013	Desde: 08:30 Hasta: 09:30	Pendiente	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Centro Cívico Enrique Tierno Galván	Star Trip (Producciones Yllana)	19/02/2013	Desde: 07:00 Hasta: 08:00	Rechazado	<input type="checkbox"/>	<input type="checkbox"/>

Página 1 de 1

GEEC 1.0.0

Figura 8.90. Listado de eventos del Editor. Modo edición

Independientemente del modo en que accedamos a esta pantalla, podremos filtrar los eventos listados en la misma utilizando los campos de entrada del formulario de la parte superior. Podremos realizar búsquedas por grupo (en el caso del Super-administrador únicamente), por espacio en el que se ejecutará el evento, por nombre del evento, por empresa organizadora del mismo y por estado en que se encuentren los eventos a buscar. Una vez introducidos los criterios de búsqueda pulsaremos el botón **Buscar** mostrándose únicamente los registros que cumplan los criterios introducidos, o el correspondiente mensaje de información en caso de no encontrarse ninguno. Si queremos volver al listado completo pulsaremos sobre el botón **Limpiar** para eliminar los criterios de búsqueda.

8.2.9 Histórico de eventos

Todos los usuarios del sistema, independientemente del rol que desempeñen, podrán acceder al listado de histórico de eventos. En esta ventana aparecerán los eventos de años anteriores que se han pasado al histórico automáticamente por el sistema.



Sesión

Usuario: jvidal
Grupo: Ayuntamiento de Leganés
Roles: Autorizador
[Desconectar](#)

Historico de eventos

Filtros de búsqueda

Espacio: Nombre:
Empresa:
[Buscar](#) [Limpiar](#)

Espacio	Título	Fecha	Horario	Estado
Centro Cívico Enrique Tierno Galván	The Hole (Producciones Yllana)	23/02/2000	Desde: 07:00 Hasta: 09:00	Ejecutado
Centro Cívico Enrique Tierno Galván	Pagagnini (Producciones Yllana)	20/02/2000	Desde: 08:30 Hasta: 09:30	Ejecutado
Centro Cívico Enrique Tierno Galván	Star Trip (Producciones Yllana)	19/02/2000	Desde: 07:00 Hasta: 08:00	Ejecutado

Página 1 de 1

GEEC 1.0.0

Figura 8.91. Listado de histórico de eventos

Se podrán realizar consultas de los mismos mediante el espacio en el que se ejecutaron, el nombre del evento y/o la empresa organizadora del mismo. Una vez introducidos los criterios de búsqueda pulsaremos el botón **Buscar** mostrándose únicamente los registros que cumplan los criterios introducidos, o el correspondiente mensaje de información en caso de no encontrarse ninguno. Si queremos volver al listado completo pulsaremos sobre el botón **Limpiar** para eliminar los criterios de búsqueda.



9 ANEXO D. Bibliografía

A la hora de desarrollar este proyecto ha sido necesario consultar distintas fuentes de información tanto para seleccionar las herramientas a utilizar y aprender su utilización, como para seleccionar las distintas tecnologías que permitiesen resolver la problemática del diseño de la arquitectura de desarrollo.

Entre las fuentes utilizadas se han consultado algunos libros, foros especializados, páginas oficiales de frameworks y aplicaciones, etc...

Las fuentes consultadas más significativas son:

- [1] Ho, Clarence; Rob Harrop. *Pro Spring 3*. Manning, Paul (editor); Jordan Elera, Manuel (technical reviewer). New York: Apress, 2012. 934 p. ISBN: 978-1-4302-4107-2
- [2] Walls, Craig. *Spring in Action*. Stirling, Sebastian (editor). Third Edition. New York: Manning, 2010. 426 p. ISBN: 978-1-9351-8235-1
- [3] Winch, Robert; Mularien, Peter. *Spring Security 3.1*. Iyer, Usha (editor); Panda, Susmita (technical editor). Birmingham: Packt Publishing, 2012. 456 p. ISBN: 978-1-8495-1826-0
- [4] Tepper Fisher, Paul; D. Murphy, Brian: *Spring Persistence with Hibernate*. New York: Apress, 2010. 264 p. ISBN: 978-1-4302-2632-1
- [5] Keith, Mike; Schincariol, Merrik: *Pro JPA2: Mastering de Java Persistence API*. New York: Apress, 2009. 500 p. ISBN: 978-1-4302-1956-9
- [6] Leonard, Anghel. *JSF 2 Cookbook..* Birmingham: Packt Publishing, 2010. 396 p. ISBN: 978-1-8471-9952-2
- [7] SPRING SOURCE. *Spring Community Forums* [En línea]. Disponible en web: <http://forum.springsource.org/forum.php>.
- [8] JAVA CODE GEEKS. *Java Code Geeks Forum* [En línea]. Disponible en web: <http://www.javacodegeeks.com>
- [9] PRIMEFACES. *Primefaces Community Forums* [En línea]. Disponible en web: <http://forum.primefaces.org>
- [10] ORACLE MySQL. *MySQL Developer Zone* [En línea]. Disponible en web: <http://dev.mysql.com>



También se han consultado diversas páginas web de espacios culturales, en concreto teatros, de las cuales obtener un dibujo de los distintos detalles técnicos que se necesitarían mantener, entre las cuales se encuentran:

- [1] TEATRO AUDITORIO DE ROQUETAS. *Instalaciones y servicios* [En línea]. Disponible en web: <http://www.teatroauditorioroquetas.org/instalaciones-y-servicios.html>
- [2] TEATRO CHAPÍ. *Ficha técnica: Escenario* [En línea]. Disponible en web: <http://www.teatrochapi.com/ficha/escenario.htm>
- [3] TEATRO MAESTRANZA. *Características técnicas del edificio* [En línea]. Disponible en web: http://www.teatromaestranza.com/secciones/teat/teat_edif_aforo.html
- [4] TEATRO METROPOLITANO JOSÉ GUTIERREZ GÓMEZ. *Especificaciones técnicas* [En línea]. Disponible en web: http://www.teatrometropolitano.com/index.php?option=com_zoo&view=item&layout=item&Itemid=115#.UMcXj4OZSDI
- [5] TEATRO MONTESSORI. *Especificaciones técnicas* [En línea]. Disponible en web: <http://www.teatromontessori.org/especificaciones.htm>